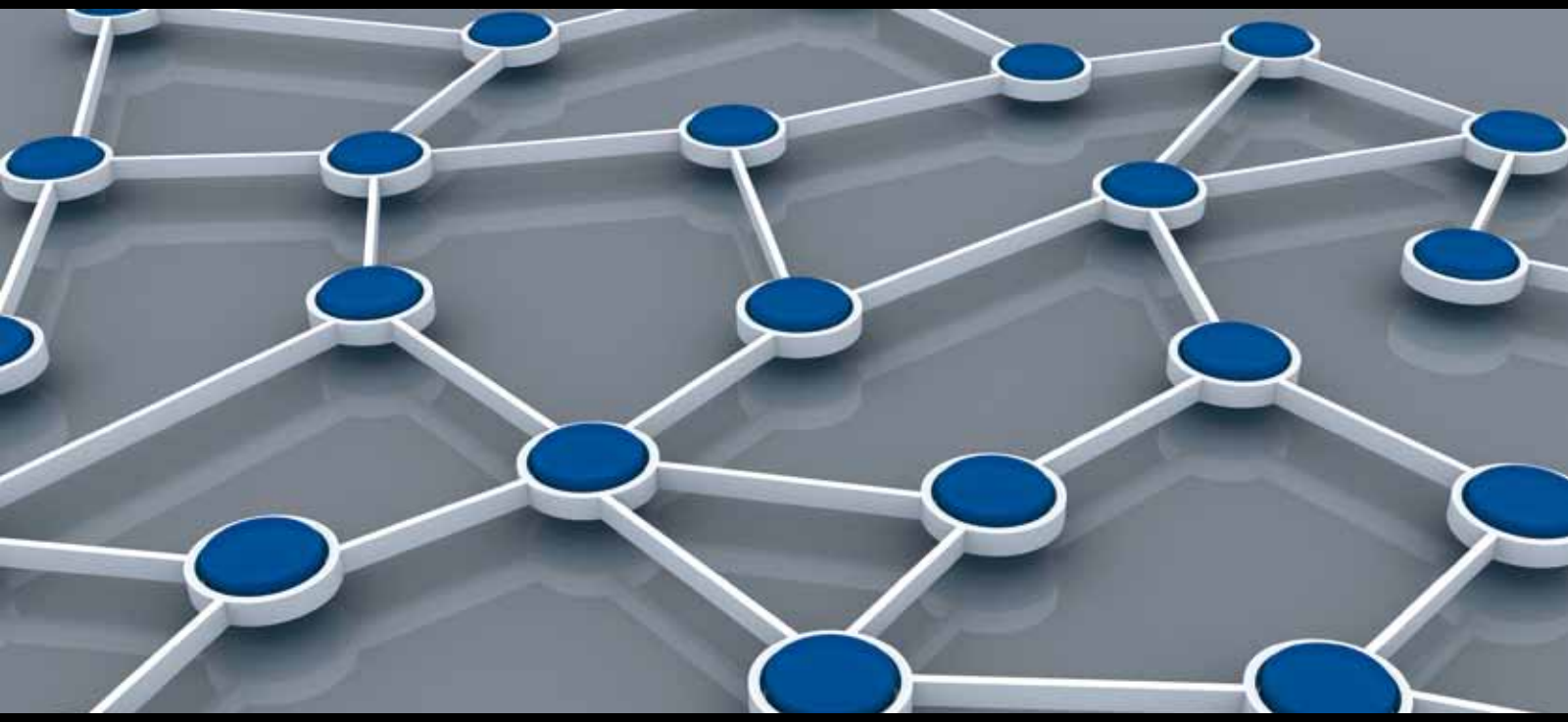
International Journal of Distributed Sensor Networks

Localization in Wireless Sensor Network

Guest Editors: Long Cheng, Carsten Maple, Chengdong Wu, and Wei Meng



Localization in Wireless Sensor Network

International Journal of Distributed Sensor Networks

Localization in Wireless Sensor Network

Guest Editors: Long Cheng, Carsten Maple, Chengdong Wu, and Wei Meng

Editorial Board

Contents

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 457874, 3 pages
http://dx.doi.org/10.1155/2013/457874

Editorial

Localization in Wireless Sensor Network

Long Cheng,[1] Carsten Maple,[2] Chengdong Wu,[1] and Wei Meng[3]

[1] *College of Information Science and Engineering, Northeastern University, Shenyang 110819, China*
[2] *Computer Science and Technology Department, University of Bedfordshire, Luton LU1 3JU, UK*
[3] *School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798*

Correspondence should be addressed to Long Cheng; chenglong8501@gmail.com

Received 3 September 2013; Accepted 3 September 2013

Wireless sensor networks that consist of thousands of low-cost sensor nodes have been used in many promising applications. Localization is one of the most important subjects because the location information is typically useful for coverage, deployment, routing, location service, target tracking, and rescue. Hence, location estimation is a significant technical challenge for the researchers. Obviously, there are some challenges for locating sensor nodes needed to be solved. The first challenge is the energy consumption and localization accuracy problem. The second challenge is the NLOS ranging error problem. The third challenge is localization in low beacon density.

The main objective of this special issue is to explore innovative, exciting, and fresh ideas for node location estimation algorithms and localization systems. Out of 30 submissions, 12 exceptional contributions were finally selected after several rounds of review by the invited reviewers and the guest editors.

The paper by J.-R. Jiang et al. proposes the AoA localization with RSSI differences (ALRDs) method to estimate angle of arrival (AoA) by comparing the received signal strength indicator (RSSI) values of beacon signals received from two perpendicularly oriented directional antennas installed at the same place. The experimental results showed that a sensor node can estimate its location by using only four beacon signals within 0.1 s with an average localization error of 124 cm. Hence, ALRD conserves the time and energy spent on localization. They further propose two methods, namely, *maximum-point minimum-diameter* and *maximum-point minimum-rectangle*, to reduce ALRD localization errors by gathering more beacon signals within 1 s for finding the set of estimated locations of maximum density. Such estimated locations are then averaged to obtain the final location estimation. Experimental results obtained demonstrate that the two methods can reduce the average localization error by a factor of about 29% to 89 cm. Hence, ALRD is suitable for mobile sensing and actuating applications, as it allows a sensor node to quickly localize itself with lower localization errors.

In the paper "*A survey of localization in wireless sensor network*," the authors classify the localization methods into target/source localization and node-self localization. In target localization, they mainly introduce the energy-based method. Then they investigate the node-self localization methods. Since the widespread adoption of the wireless sensor network, the localization methods are different in various applications. So there are several challenges in some special scenarios. They present a comprehensive survey of these challenges: localization in non-line-of-sight, node selection criteria for localization in energy-constrained network, scheduling the sensor node to optimize the tradeoff between localization performance and energy consumption, cooperative node localization, and localization algorithm in heterogeneous network. Finally, they introduce the evaluation criteria for localization in wireless sensor network.

The paper by P. Gao et al. proposes a path-planning, a location predicting method (PPLP) for indoor mobile target localization. They firstly establish the path-planning model to constrain the movement trajectory of the mobile target in indoor environment according to indoor architectural pattern. Then, they use MLE approach to get one certain location result of the target. After that, based on the path-planning model and some previous localization results of the target, the best possible position of the target in the next

time interval can be predicted with the proposed predicting approach. Finally, the MLE result and prediction result are weighted to obtain the final position.

In the paper "*Localization with single stationary anchor for mobile node in wireless sensor networks*," the authors propose a localization algorithm named LSARSSI for mobile node based on received signal strength indicator (RSSI) between locating sensor node with inertia module built-in and the single anchor. In order to avoid errors from directly mapping absolute RSSI values to distances, they obtain the geometrical relationship of sensors by contrasting the measured RSSI values. They then design a novel localization scheme, LSARSSI, which has a better accuracy and low overhead. The simulation results show that the proposed schemes perform high accuracy and feasibility, even in large-scale environment.

In the paper "*Range-free localization scheme in wireless sensor networks based on bilateration*," the authors propose a low-cost yet effective localization scheme for wireless sensor networks (WSNs). The proposed scheme uses only two anchor nodes and uses bilateration to estimate the coordinates of unknown nodes. In this scheme, two anchor nodes are installed at the bottom-left corner (Sink X) and the bottom-right corner (Sink Y) of a square monitored region of the WSN. Sensors are identified with the same minimum hop counts pair to Sink X and Sink Y to form a zone, and the estimated location of each unknown sensor is adjusted according to its relative position in the zone. Simulation results show that the proposed scheme outperforms the DV-Hop method in localization accuracy, communication cost, and computational complexity.

The paper by Y. Wang et al. proposes a prior knowledge-based correction strategy (PKCS) to locate the robot. They firstly investigate the RSS-based NLOS identification method using the recorded measurements. Then the ratio of NLOS present in the record of measurements and the expectation of the NLOS errors are used to mitigate the NLOS errors. Kalman filter is employed to improve the estimated range. Finally, they use the residual weighting algorithm to estimate the location of the robot. Simulation results show that the PKCS has much better performance than those methods without the correction method and significantly improves the localization accuracy.

The paper by S. Zhang et al. proposes a human motion tracking approach for daily life surveillance in a distributed wireless sensor network using ultrasonic range sensors. It uses cheap range sensor nodes in wireless sensor networks by jointly selecting the next tasking sensor and determining the sampling interval based on predicted tracking accuracy and tracking cost under the UKF frame. Simulation results show that the new scheme can achieve significant energy efficiency without degrading the tracking accuracy.

In the paper "*On the joint time synchronization and source localization using toa measurements*," the authors consider the problem of estimating the clock bias and the position of an unknown source using time of arrival (TOA) measurements obtained at a sensor array to achieve time synchronization and source localization. The mean square error (MSE) analysis is firstly performed for the case where the source is localized via TOA positioning when assuming the source clock bias does not exist, but in fact it is non-zero. Comparing the obtained source localization MSE with that from joint estimating the source position and clock bias, they derive a condition under which ignoring the source clock bias may provide a smaller localization MSE. Computer simulations are conducted to corroborate the theoretical development and illustrate the good performance of the proposed algorithm.

The paper by D. Arbula and K. Lenac presents Pymote, the library that provides support for simulation and analysis of distributed algorithms built on top of comprehensive Python environment. Pymote is designed to allow rapid interactive testing of new algorithms, their analysis, and visualization while minimizing developer's time. It supports both interactive algorithm simulation and automation of experiments and provides visualization tools for both. It has been deliberately kept simple, easy to use, and extensible.

The paper by E. Navarro-Alvarez et al. presents a new adaptive method to calculate the path loss exponent (PLE) for microcell outdoor dynamic environments in the 2.4 GHz industrial, scientific, and medical (ISM) frequency band. The main contribution of this method is the formulation of a parametric mathematical model which improves the PLE accuracy by using the *equivalent isotropic radiated power* (EIRP) and *effective antenna aperture* (EAA) parameters calculated before obtaining the PLE. A second contribution is the combination of GPS data and RSSI readings in order to identify the RSSI long term behavior.

In the paper "*A novel lightness localization algorithm based on anchor nodes equilateral triangle layout in WSNs*," the authors present a novel equilateral triangle localization algorithm (LETLA) that is a lightweight approximate localization algorithm and could provide better precision with less power consumption. The LETLA is an approximate localization based on the concept of substituting the approximate coordinates for the real coordinates, which could result in less accuracy and save more energy. In order to avoid the ranging ambiguities arising from the interference of noise, the LETLA adopts the order of ranging results to represent the location relationship of unknown node and anchors. Simulations show that the LETLA performs better than other state-of-the-art approaches in terms of energy consumption with the same localization precision.

The paper by X. Qu et al. presents minimax estimation fusion method in distributed multisensor systems. This method aims to minimize the worst-case squared estimation error when the cross-covariances between local sensors are unknown and the normalized estimation errors of local sensors are norm bounded. The simulation results illustrate that the proposed fusion method is a robust fusion in localization and tracking and more accurate than the previous covariance intersection method.

Acknowledgment

The guest editors would like to acknowledge the contributions of the many experts who submitted their works.

We thank all the reviewers for their helpful feedback and suggestions to improve the quality of each paper.

Long Cheng
Carsten Maple
Chengdong Wu
Wei Meng

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 912029, 10 pages
http://dx.doi.org/10.1155/2013/912029

Research Article

GPS-Assisted Path Loss Exponent Estimation for Positioning in IEEE 802.11 Networks

Ernesto Navarro-Alvarez,[1] Mario Siller,[1] and Kyle O'Keefe[2]

[1] *Electrical Engineering and Computer Science Department, CINVESTAV-Unidad, Guadalajara. Av. Del Bosque 1145, Colonia El Bajio, Zapopan, JAL 45019, Mexico*
[2] *Geomatics Engineering Department, University of Calgary, Schulich School of Engineering, 2500 University Dr. NW, Calgary, AB, Canada T2N 1N4*

Correspondence should be addressed to Ernesto Navarro-Alvarez; enavarro@gdl.cinvestav.mx

Received 3 November 2012; Revised 2 May 2013; Accepted 8 May 2013

Academic Editor: Chenglong Cheng

We present a new adaptive method to calculate the path loss exponent (PLE) for microcell outdoor dynamic environments in the 2.4 GHz Industrial, Scientific, and Medical (ISM) frequency band. The proposed method calculates the PLE during random walks by recording signal strength measurements from Radio Frequency (RF) transceivers and position data with a consumer-grade GPS receiver. The novelty of this work lies in the formulation of signal propagation conditions as a parametric observation model in order to estimate first the PLE and then the distance from the received RF signals using nonlinear least squares. GPS data is used to identify long term fading from the received signal's power and helps to refine the power-distance model. Ray tracing geometries for urban canyon (direct line of sight) and nonurban canyon (obstacles) propagation scenarios are used as the physics of the model (design matrix). Although the method was implemented for a lightweight localization algorithm for the 802.11b/g (Wi-Fi) standard, it can also be applied to other ISM band protocols such as 802.15.4 (Zigbee) and 802.15.1 (Bluetooth).

1. Introduction

The emergence of context-aware computing applications, and location-based services and the proliferation of portable electronic devices have motivated an extensive research on the topic of node or device localization for wireless networks operating under 2.4 GHz ISM band protocols such as Bluetooth, Zigbee, and especially Wi-Fi. Localization is an important issue for the interaction between portable device users and the surrounding networked devices in intelligent environments such as homes, offices, or other intelligent buildings [1, 2].

The localization problem has been a hot research topic in the Wireless Sensor Network (WSN) and Mobile Robotics literature. According to the WSN terminology, a *node* is a small device with sensing, computing, storage, and communication capabilities, and the *node localization* problem is defined as "determining an assignment of coordinates for nodes in a wireless ad-hoc or sensor network that is consistent with measured pair wise node distances" [3]. The definition states basically that it is necessary to estimate a distance (range) between the nodes before obtaining coordinates. See [4–6] for more WSN terminology and applications.

A range estimation can be performed using collected measurements from a variety of methods such as acoustic [7–9], directional antenna or antenna array [10, 11], infrared [12], and Received Signal Strength (RSS) measurements [13, 14]. Unlike other ranging methods, RSS-based methods estimate neither distances nor positions from the angle of arrival nor traveling time of the signal. Additionally, no clock synchronization is assumed in the nodes. Only the incoming RF signal is used to infer a range to the transmitter. The basic premise is that the received signal power decay is inversely proportional to the distance. This signal power attenuation is called path loss (PL) and can be quantified by a path loss exponent (PLE). Once the ranges to the landmarks (devices with known positions) are estimated, trilateration methods can be employed to find the location of an unknown position node within a suitable coordinate system.

However, RSS levels are highly unpredictable and the formulation of a path loss model (PLM) as function of distance is a complex issue. This complexity is derived from the fact that the received power level is a combination of different signal propagation mechanisms such as reflection, diffraction, and scattering and absorption losses. Due to these impairments, which depend on the surrounding environment, *shadowing* and *multipath fading* are produced. As a consequence estimation errors can be introduced for any RSS-based localization algorithm.

In recent years, RSS-based localization algorithms have been the subject of an increasing interest due to the wide availability of 802.11b/g transceivers and the proliferation of Wireless Local Area Networks (WLANs).

This approach exploits existing WLAN infrastructure and precludes the use of additional hardware such as badges; and wearable sensors, see [15, 16]. In fact, the IEEE 802.11 standard provides the means to obtain the signal strength via the Received Signal Strength Indicator (RSSI). This indicator is defined as "a mechanism by which RF energy is to be measured by the circuitry on a wireless NIC. This numeric value is an integer with an allowable range of 0–255 (a 1-byte value)" [17]. Similar metrics are defined for the IEEE 802.15.4 and IEEE 802.15.1 standards.

The node localization algorithms within previous literature are referred to as GPS-free or GPS-less algorithms. The lack of GPS use can be explained by common drawbacks attributed to several causes such as signal availability, cost, antenna size, and energy consumption. Nowadays, some of these arguments are no longer true. GPS chips can be ubiquitously found in mobile devices such as smart phones, hand helds, and tablets. While GPS chips are available in many devices, there are still many others that do not have it. The existing GPS positioning capability could be used to estimate the PLEs of WLAN access points and the derived PLM employed to enable GPS-free location services.

In this paper we propose a PLE estimation method which uses the RSSI provided by a 802.11b/g transceiver in combination with data collected from a commercial grade GPS receiver. The method builds ray tracing models for typical propagation scenarios such as *urban canyon* and *non-urban canyon* cases, and uses them to formulate the design matrix of an observation model. The propagation scenarios take into account the shadowing and multipath effects. The observations are composed of RSSI readings and GPS data. The system of equations of the design matrix is linearized using Taylor series and then solved through least squares. The main contribution of this work is the formulation of a parametric mathematical model which improves the PLE accuracy by using the *Equivalent Isotropic Radiated Power* (EIRP) and *Effective Antenna Aperture* (EAA) parameters calculated before obtaining the PLE. A second contribution is the combination of GPS data and RSSI readings in order to identify the RSSI long term behavior.

Depending on the transmitter's power, antenna height, and coverage area, the RF environments where the devices are deployed can be classified as macrocell, microcell, and picocell. In microcell environments the transmitting power of the radios ranges from 0.1 to 1 watt, the RF coverage area ranges from 200 to 1000 m, and the transmitter's height is low (3 to 10 meters) [18]. Environments within buildings are classified as picocells. Indoor-to-outdoor configurations are environments with walls blocking the signal and are characterized by a wall attenuation factor. The algorithm proposed in this work is tailored to microcell RF environments with indoor-to-outdoor coverage configurations; therefore models such as Okumura-Hata, Lee, or Walfisch-Bertoni are not treated here. From this point forward, when we refer to a blind node, we mean a device (smart phone, hand held, tablet, or laptop) whose position needs to be estimated; when we refer to anchor or beacon or landmark nodes, we mean access points (APs) for whose position are already known.

The paper is organized as follows. Section 2 reviews the localization algorithms based on RSSI only and algorithms with RSSI-GPS collaboration. In Section 3 we present the proposed method. Section 4 presents the implantation of the method in real world conditions. In Section 5 the accuracy of the experimental results is discussed, and finally, conclusions and future work are presented in Section 6.

2. Related Work

This section is divided in two parts. The first part reviews papers related to either empirical or theoretical RSSI-based models. In the second part, techniques that rely on both RSSI and GPS to derive power-distance models for node localization are examined. Note there are other methods for RF-based localization such as fingerprinting [19, 20] and Bayesian Networks [21, 22]. These techniques and some others exclude PLE estimation and are not reviewed in this paper.

2.1. PLE from RSSI Measurements. The study of RSSI for different purposes is not a new idea. Some of these purposes are the optimization of the coverage area of wireless networks [23], assessment of links quality for multihop routing protocols [24], and so forth. The use of RSSI as a means for node localization estimation in a WLAN dates back to 2000. For example, [25] proposes the use of an Extended Kalman Filter (EKF) to cope with the noise in the measurements and to maintain a position estimate in harsh conditions. They conducted an empirical experiment to relate signal strength to distance between base and mobile stations. The shortcoming of this work is that it is designed to work within predefined surroundings, in this case an office. The accuracy is one room. In [26] authors propose a lookup-table method for node position triangulation. They also conducted data collection to empirically correlate distance to signal strength. The shortcoming of this works is that it requires an extensive survey at different points in a predefined environment and the consequent table size. In [27] authors proposed an optimal averaging window length of RSSI samples to cope with fading of the power and mobility of the nodes. By modeling the channel's fading with a Rayleigh distribution, they obtained a factor (r^2) which multiplies the RSSI samples (w). The authors reported a lower bound mean error of 2.5 m with $w = 50$. The main drawback is that this method is intended for rectangular areas where bacons are placed optimally.

Although these previous works do not calculate a PLE explicitly, they formulate empirical power-distance models based on averaged RSSI measurements surveyed in specific locations. They also address some issues affecting the received power levels such as fading due to multipath, nonline of sight, node mobility, and sampling.

In the following literature review we now focus on localization systems for outdoor environments since PLE estimation was first introduced for such environments. In [28] the authors propose a methodology for PLE estimation in a WiMAX system. Although this methodology is not intended for a WLAN, it is carefully examined in this paper because of the use of common theoretical models to estimate PLE. The first part of the methodology pairs RSSI measurements, expressed in dB units, with distance (L[dB], d[m]). In the second part, the authors use the well-known log-distance path loss model to formulate a system of equations. The authors measured propagation losses in two different points with identical conditions (line of sight condition) and formulated a system of two linear equations:

$$\begin{aligned} L_1\,[\mathrm{dB}] &= a + 10 * \gamma * \log_{10}(d_1), \\ L_2\,[\mathrm{dB}] &= a + 10 * \gamma * \log_{10}(d_2), \end{aligned} \tag{1}$$

where a is a coefficient that accounts for frequency and other propagation factors, γ is the PLE, and d is distance in meters between the transmitter and the receiver. Solving the system and using several measurements the authors obtained the following empirical path loss model L[dB] $= 123.02 + 10 * 2.687 * \log_{10}(d)$.

In [29] the authors used the Okumura-Hata model to calculate two different PLEs in a WiMAX network, although the Okamura-Hata model is usually applied in macrocell environments (distances greater than 1 km). They formulated two different map-supported PLMs from RSSI observations. One PLM corresponds to an "urban canyon" area while the other for "non-urban canyon." The developed models considered the type of area based on city map and road network information.

Note that an urban canyon area is an area where there exists an open street between the receiver and the transmitter. Therefore, the signal reaches longer distances than in areas with obstacles. On the other hand, an area with considerable obstacles is classified as noncanyon.

Next, we shall review works which estimate PLE for WLAN in indoor environments. Authors in [30] proposed and implemented a system framework which consists of a central server, a base station, and four beacon nodes. The algorithm dynamically estimates a PLE between the beacons and the blind node. The base station receives the RSS values collected by the beacons nodes and sends them to the server. The authors also employed the log-distance path loss formula, but they add a stochastic component:

$$\mathrm{PL}(d) = \mathrm{PL}(d_0) - 10\alpha\log_{10}\left(\frac{d}{d_0}\right) + X_\sigma, \tag{2}$$

where PL(d) denotes the path loss in dB as function of distance d, in meters, away from sender; PL(d_0) is a path loss constant at a reference distance d_0; α is the PLE; and X_σ is Gaussian noise in dBm units. The X_σ values account for the long term variability. The α exponent is estimated using the following formula:

$$\alpha = \frac{-\sum_{i=1}^{n}\left(\mathrm{PL}(d_i) - \mathrm{PL}(d_0)\right)}{\sum_{i=1}^{n} 10\left(\log d_i\right)}, \tag{3}$$

where n is the number of RSSI measurements at distances d_i. Basically, (3) expresses α in terms of an averaged ratio between path loss and the logarithm of distance using all measurements recorded. This averaged PLE is used to obtain distances between the blind node and the four beacons. Finally, a polygon method was used to obtain the blind node's coordinates.

Model (2) is commonly used in macrocell scenarios by setting the d_0 reference distance to 1 km. Although the formula was modified for microcell scenarios (setting d_0 to 1 and 100 meters), we do not think that this is the most suitable PLM because of the particularities of such environments.

2.2. RSSI Measurements and GPS Data. Some of the first applications of collaborative GPS/Wi-Fi were geocaching, wardriving, and the elaboration of signal coverage maps. Geocaching is a recreational outdoor activity in which the users seek containers with the aid of GPS receiver and mobile devices. War driving is the activity of searching Wi-Fi wireless networks in a moving vehicle, using a portable computer or other devices connected to a GPS. A signal coverage map is a map with geographic information of areas where a wireless networks are deployed. It represents signal intensity areas (strong or weak) with contour lines or colors.

In [31] the authors presented a solution for manual deployed networks called Walking GPS. It works by attaching a GPS receiver to a node called GPS mote. This node first converts its latitude and longitude coordinates into a local coordinates system and then it broadcasts its position to the rest of the nodes. When the carrier (person or vehicle) places a new node in a certain position and turns it on, the node immediately receives the broadcast packet from the GPS mote and estimates its own position. On the other hand, if a node is turning on after being deployed, it needs to ask its neighbors for their positions in order to trilaterate its own position. The main drawback of this solution is that it was tested in an ideal propagation scenario (open field environment). Moreover, the blind nodes were deployed within a predefined grid.

In [32] the authors proposed a multisensor fusion solution with data from three sources: a GPS, a radio propagation map, and a WLAN positioning system. They divided the radio map into three areas: indoor, outdoor, and shaded (a shaded area is an area surrounded by buildings or in closed places). The areas are covered by three fixed APs. The algorithm consists of two phases: offline and online. In the offline phase they collect RSSI measurements at predefined locations in the map. At these locations, they estimate the expected RSSI values using an equation similar to (3) with a reference distance of 1 meter. Calculating the ratio between the observed and the expected RSSI, they model the corresponding multipath. Based on these ratios, a polynomial fit function is computed

for all the predefined locations. Using this information and trilateration, the WLAN positioning system estimates a blind node preliminary position. Finally, in the online phase the GPS is used to identify the area (indoor, outdoor, or shaded) to improve the estimated positions.

In [33] the authors presented an outdoor WiFi localization system assisted by GPS. The system uses a unidirectional Yagi type antenna to triangulate the location of APs using the angle of arrival of the received signal. The angle of the received signal is measured with a GPS compass which is rotated with the antenna and the WiFi receiver at the same time. All the equipment was mounted on a motorized rotating base. The proposed algorithm consists of the following steps: (i) place the equipment at two different measurements points $M_1(x_1, y_1)$ and $M_2(x_2, y_2)$; (ii) find the respective θ_1 and θ_2 which corresponds to the angles at which the maximum RSSI are observed; (iii) find the slopes $m_1 = \tan\theta_1$, and $m_2 = \tan\theta_2$ and calculate c_1 and c_2 using the line equation $y = mx + c$; and (iv) find the intersection point for $y_1 = m_1x_2 + c_1$ and $y_2 = m_2x_2 + c_2$. This intersection point corresponds to the estimated location of the AP.

3. Proposed Solution

The models described in Section 2 express a power distance relationship based on the attenuation of the signal's power as it propagates. Such attenuation roughly obeys the inverse power law:

$$P_r = \frac{1}{d^{\alpha}}, \tag{4}$$

where P_r is the power received in watts, α is the PLE (equal to 2 in free space conditions), and d is the distance between the transmitter and the receiver in meters. However, the receiver power attenuates at a much higher rate and exponent $\alpha > 2$. A higher α can be explained in terms of losses caused by propagation mechanisms, such as *diffraction*, *scattering*, *reflection*, and *refraction*. For a detailed explanation refer to [34]. The combination of these mechanisms is responsible for power variations in the RSSI readings. These variations are commonly classified as *slow variation* or *long term fading* and *fast variations* or *short term fading*. Fast variations are characterized by rapid fluctuations in the RSSI levels over very short distances. On the other hand, *long term fading*, also called large scale path loss, is due the increasing distance as the receiver moves away from the transmitter.

A suitable mathematical tool to model these propagation mechanisms and their effects on RSSI variability is ray-tracing. An electromagnetic wave (EM) is composed of electric (**E**) and magnetic (**B**) fields. These fields are perpendicular to each other and the direction of the wave is obtained from the cross product of $\mathbf{E} \times \mathbf{B}$. The result is the Poynting vector (**S**) that can be modeled as a ray. In ray tracing, a ray is an imaginary straight line depicting the path light travels. Using geometrically defined propagation scenarios, the trajectory of different **S** rays can be computed.

The proposed method is divided into three phases. In the first phase the EIRP and PLE parameters are estimated using the observation model. In the second phase, PLMs for each landmark are formulated in order to translate RSSI measurements into ranges. In the final phase, these ranges are used to estimate the locations of one or more blind nodes. Next, each phase is described in detail.

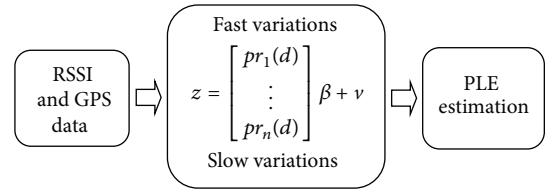


Figure 1: Schematic diagram for phase 1.

3.1. Observation Model Formulation. The formulation of the system to infer the EIRP and PLE is expressed as follows:

$$\mathbf{z} = H(\mathbf{x})\,\beta + \mathbf{v}, \tag{5}$$

where $\mathbf{z} = [z_1 z_2 z_3 \cdots z_n]$ is a $1 \times n$ vector of RSSI observations recorded in dB units at different distances ranging from 1 to 200 meters. Each z_i represents the measurements at a particular distance (see Section 4); $H(\mathbf{x})$ is the design matrix representing the ray tracing models; $\beta = [\theta_1, \theta_2]$ is the vector of parameters to be estimated; and lastly **v** is a Rayleigh distributed random variable which accounts for fast variation in short distances [35]. The block diagram for this phase is shown in Figure 1.

Mathematical models expressed in matrix H relate the observations **z** with the parameter vector β. These models are ray tracing equations for an urban canyon and a non urban canyon. Matlab scripts provided by [36] are used to construct the geometry for the two scenarios. See Figure 2.

The models perform the addition of direct and reflected rays and calculate the resulting received power at different points in the scenarios. This is done by calculating the average of vector **S** trough an area, that is, the *power flux density* (PFD). First average **S** is expressed in terms of **E**-field strength and then it is related to the *Effective Antenna Aperture* (EAA) area of the receiving antenna as follows:

$$P_r = \emptyset A_r. \tag{6}$$

The PFD represents the field strength at the receiver's antenna (in W/m^2 units) and it is defined as

$$\emptyset = \frac{1}{2}\frac{|e|^2}{120\pi}, \tag{7}$$

where $|e|$ is the magnitude of the electric field radiated in the far-field region by the source point. 120π is the impedance of free space (in Ohms). EAA is defined as

$$A_r = \frac{\lambda^2 \cdot g_r}{4\pi \cdot l_r}, \tag{8}$$

where g_r is the receiving antenna gain in dBi, and l_r is the system loss at the receiver. PFD is defined as follows:

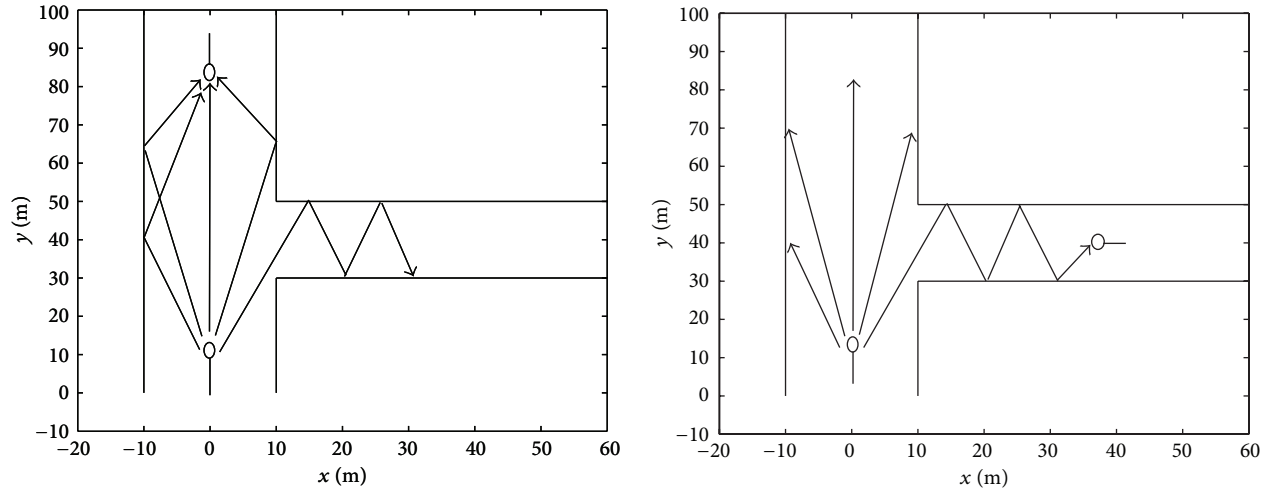


Figure 2: Ray tracing representation of signals traveling distances in (a) urban canyon propagation scenario and (b) nonurban Canyon propagation scenario.

Expressing (6) as a function of distance

$$P_r(d_0) = A_r \frac{|e_T(d_0)|^2}{2\eta}, \tag{9}$$

where e_T is the modulus of complex number e_T and represents the total field strength of the rays combining at the receiver ($e_T = e_0 + e_1 + e_2 + e_3$). The term e_0 represents the direct ray field strength; e_1 represents the ray reflected in the ground; and e_2, e_3 represent the rays reflected on walls. The formulas for each term are

$$\begin{aligned} e_0 &= \sqrt{\frac{60pg}{l}} \left\{ \frac{e^{(-jkd_0)}}{d_0} \right\}, \\ e_1 &= \sqrt{\frac{60pg}{l}} \left\{ \mathrm{GR} \frac{e^{-jk(d_0+d_1)}}{d_0+d_1} \right\}, \\ e_2 &= \sqrt{\frac{60pg}{l}} \left\{ \mathrm{WR} \frac{e^{-jk(d_0+d_2)}}{d_0+d_2} \right\}, \\ e_3 &= \sqrt{\frac{60pg}{l}} \left\{ \mathrm{WR} \frac{e^{-jk(d_0+d_3)}}{d_0+d_3} \right\}, \end{aligned} \tag{10}$$

where d_0 is the direct ray distance between the transmitter and the receiver, d_1 is the additional distance the ray travels due to reflection on the ground, d_2 and d_3 are the additional distances due to wall reflections. These additional distances (d_1, d_2, d_3) are fixed according to the specified geometry. Constants WG and WR are ground and wall reflections coefficients, respectively. Substituting (10) in (9) we obtain:

$$A_r \left| \sqrt{\frac{60pg}{l}} \left(\left\{ \frac{e^{(-jkd_0)}}{d_0} + \mathrm{GR} \frac{e^{-jk(d_0+d_1)}}{d_0+d_1} + \mathrm{WR} \frac{e^{-jk(d_0+d_2)}}{d_0+d_2} + WR \frac{e^{-jk(d_0+d_3)}}{d_0+d_3} \right\} \right) \right|^2 (2\eta)^{-1}. \tag{11}$$

The term $p_t g_t / l_t$ is the EIRP and one of the parameters to estimate. Substituting (11) in matrix H_A in (5), the resulting system is

$$\mathbf{z} = \begin{bmatrix} Pr_1(d) \\ \vdots \\ Pr_n(d) \end{bmatrix} \beta + \mathbf{v}. \tag{12}$$

3.2. Range Estimation. After linearizing (12) using Taylor series, we apply least squares to estimate EIRP and EAA. In order to obtain a more accurate PLE, we use transmitted power, p_t received power, p_r transmitter antenna gain, g_t receiver antenna gain, g_r, transmitter system loss, l_t and receiver system loss, l_r, to formulate a new system:

$$\mathbf{z} = \begin{bmatrix} Pr_1(d) \\ \vdots \\ Pr_n(d) \end{bmatrix} \beta + \mathbf{n}. \tag{13}$$

In this formulation, the RSSI observation vector $\mathbf{z}$ contains the short term or slow variations which were filtered

Table 1

AP ssid	Actual coordinates	Estimated PLE (γ)
ENA_131	N: 5662636.17 E: 700932.53	2.42
ENE_136	N: 5662669.42 E: 700841.48	2.35
NM_110	N: 5662542.19 E: 700931.53	2.31
KNB_125u	N: 5662445.24 E: 700805.63	2.40
KNB_132	N: 5662421.01 E: 700798.60	2.41
MC_184	N: 5662452.13 E: 700879.63	2.52
MC_197	N: 5662474.47 E: 700927.79	2.55

out of the raw readings. This extraction was carried out by a running mean. A running mean calculates the signal strength averages within a certain length distance interval. Commonly used length interval ranges from 20 to 40 times λ [37]. These averages are indexed and the vector $\mathbf{z}$ is formed where $n = 1 \cdots \max\_distance$. The variable $\mathbf{n}$ is a normal distributed random variable which accounts for slow variations in long term fading. The model for this system's design matrix is the logarithm of the Friis formula:

$$p_r(d) = (p_t + g_t + g_r - l_t - l_r) + \left(10 \cdot 2 \cdot \log_{10}\left(\frac{\lambda}{4\pi}\right)\right) + \gamma \cdot 10 \cdot \log_{10}(d). \tag{14}$$

The terms between parentheses are constant and were calculated in the previous step; therefore the parameter, $\beta = [\theta]$, to be estimated is the PLE, γ, and remains in the last term.

Seven APs located in the University of Calgary campus were selected as landmarks during the data collection process (Refer to Section 5 for more information) having the following service set identification (ssid): ENA_131, ENE_136 and NM_110, KNB_125u, KNB_132, MC_184, and MC_197. After the method was applied to the signal strength received from them and to the GPS data collected, a PLE and a path loss model for each one were estimated. Table 1 lists ssid, estimated PLE values, and the actual UTM coordinates for the landmarks. In this work the UTM system is employed due to its use of meters instead of degrees of latitude and longitude.

Figure 3 shows short term received power and the corresponding long term path loss model for APs: ENA_131, ENE_136, and NM_110. Figures 3(c) and 3(d) correspond to non-urban canyon and reflect more harsh propagation conditions.

3.3. Node Localization. In order to determine the blind node position in 2D space, n landmarks are employed. Although two landmarks would be enough, there would be two possible solutions. A third landmark reduces the estimation to a unique solution. With the ranges estimated from a blind node to n landmark APs and the coordinates of the latter, the method can formulate the following system of nonlinear simultaneous equations:

$$\begin{aligned} \rho_1 &= \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2}, \\ \rho_2 &= \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2}, \\ &\vdots \\ \rho_n &= \sqrt{(x_n - x_u)^2 + (y_n - y_u)^2 + (z_n - z_u)^2}, \end{aligned} \tag{15}$$

where (x_1, y_1), (x_2, y_2), and (x_3, y_3) are the APs known coordinates; (x_u, y_u) is the position to find; and $\rho_1, \rho_2, \ldots, \rho_n$ are the estimated ranges. UTM Northing and Easting coordinates are identified with y and x, respectively. Because the blind node and the APs are placed at ground level, the z coordinate is constant with an altitude of 1112.4 meters. Because the number of equations is larger than the number of unknowns the system has no analytical solution. However, it can be solved through linearization and iteration. First we decompose the unknowns into approximate and incremental components: $x_u = \widehat{x}_u + \Delta x_u$, $y_u = \widehat{y}_u + \Delta y_u$, $z_u = \widehat{z}_u + \Delta z_u$, where Δx_u, Δy_u, Δz_u are the unknowns, and $\widehat{x}_u$, $\widehat{y}_u$, $\widehat{z}_u$ are considered known by assigning them an initial value. By substituting these values in (15) and differentiationing we obtain:

$$\begin{aligned} &p(\widehat{x}_u + \Delta x_u, \widehat{y}_u + \Delta y_u, \widehat{z}_u + \Delta z_u) \\ &\quad = p(\widehat{x}_u, \widehat{y}_u, \widehat{z}_u) + \frac{\partial p(\widehat{x}_u, \widehat{y}_u, \widehat{z}_u)}{\partial \widehat{x}_u} \Delta x_u \\ &\qquad + \frac{\partial p(\widehat{x}_u, \widehat{y}_u, \widehat{z}_u)}{\partial \widehat{y}_u} \Delta y_u + \frac{\partial p(\widehat{x}_u, \widehat{y}_u, \widehat{z}_u)}{\partial \widehat{x}_u} \Delta z_u + \cdots. \end{aligned} \tag{16}$$

The series is truncated after the first-order partial derivatives eliminating nonlinear terms. With initial values for $\widehat{x}_u$, $\widehat{y}_u$, $\widehat{z}_u$, new values for Δx_u, Δy_u, Δz_u can be calculated and used to modify original $\widehat{x}_u$, $\widehat{y}_u$, $\widehat{z}_u$ values. The modified values are used again to find new deltas. This iteration continues until the absolute values of deltas are within a certain predetermined limit. For a detailed explanation of this process see [38].

For instance, with Wi-Fi only data collected at test point 5662622.36N, 700894.86E, and the PLEs and coordinates of APs ENA_131, ENE_136, and NM_110 (listed in Table 1), the following mean ranges were obtained: $\rho_1 = 43.9158$ meters, $\rho_2 = 60.5400$ meters, and $\rho_3 = 65.5568$ meters. Real ranges are: 38.58, 68.42, and 91.43 meters. After solving the system, the resulting estimated coordinates for the blind node were: $x_u = 5662600$, $y_u = 700900$, and $z_u = 1112.4$ (red color balloon in Figure 4). In this particular case the position error was 22.9432 m.

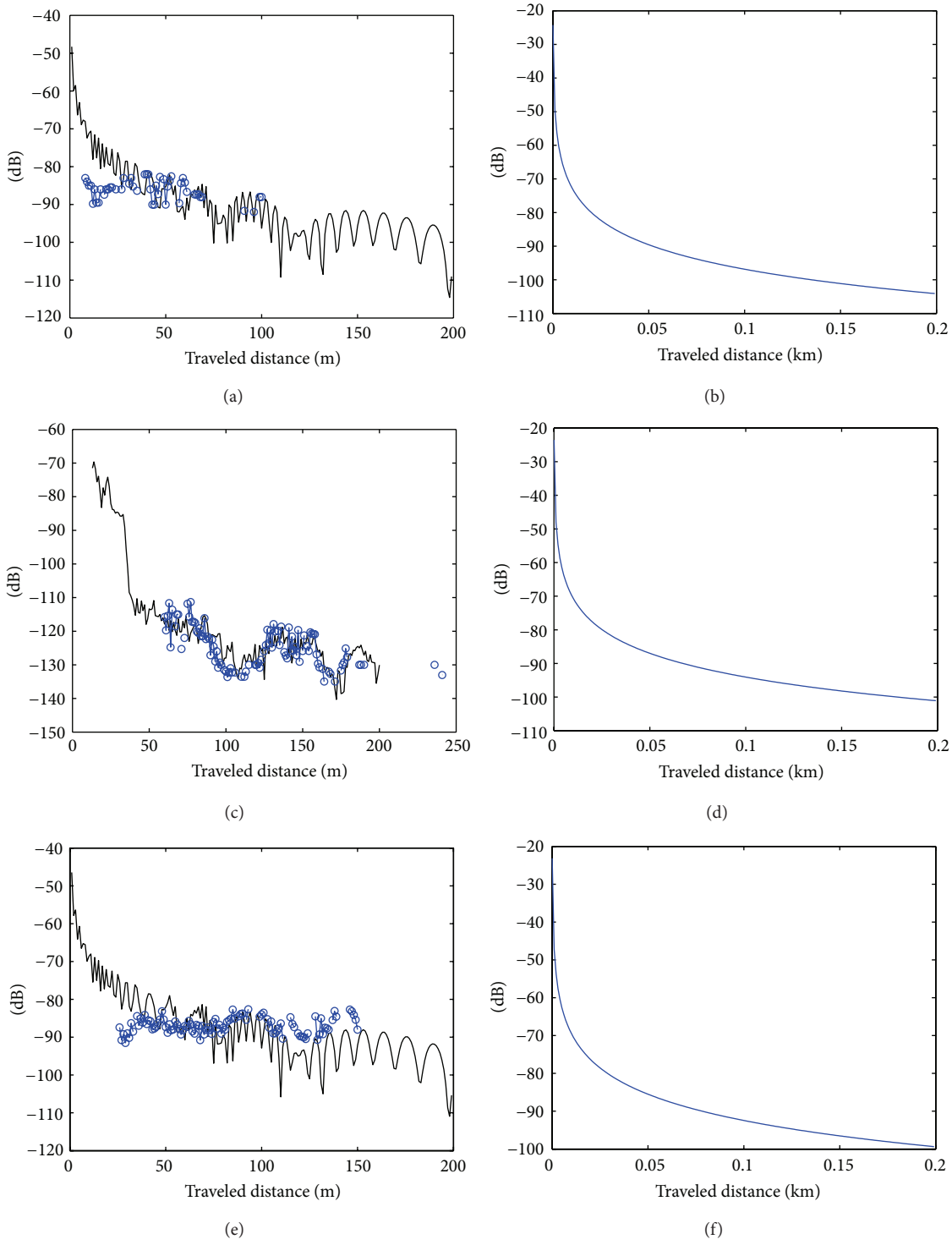


Figure 3: (a) AP ENA_131 short term received signal power for an urban canyon model. (b) Corresponding long term distance model with PLE = 2.42. (c) AP ENE_136 short term received signal power for non-urban canyon. (d) Corresponding long term distance model with PLE = 2.35. (e) AP NM_110 short term received signal power for urban canyon model. (f) Corresponding long term model with PLE = 2.31.

4. Implementation of the Method

In order to test our proposal, several experiments were performed at a university campus. The data collection process consists in recording the signal strength from nearby Wi-Fi APs. The geographic location of the points where data was collected was also logged and paired with the Wi-Fi data at rate of 3 Hz. The data collection was carried out with a laptop

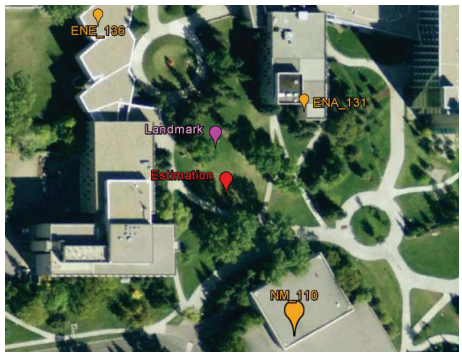


Figure 4: One example result and visualization on Google Earth.

and a GPS Garmin model GPSmap 60CSx. In the laptop the inSSIDer program was installed and the GPS receiver was connected via USB port. The campus infrastructure consists of 1280 APs with transmission rate of 54 Mbps, maximum Tx sensitivity of +17.0 dBm, and minimum Rx sensitivity of −73.0 dBm. 87 out of 1280 APs were identified along with their geographic positions, but only seven were used in the experiments (see Table 1). The data was collected during several random walks each 15 minutes long and was carried out on August 7, 11, 15, and 29 and June 15 and 23, 2011. All tests took place within the area located between the three buildings. See Figure 4. The data was logged in the GPS eXchange format:

```
<time>2011-08-16T00 : 26 : 46.0Z</time>
<wpt lat="51.079936" lon="−114.133793">
<MAC>00 : 0B : 86 : D6 : CB : 43</MAC>
<RSSI> −78</RSSI>
```

The <time> tag stores the UTC time when the measurement was taken. The <wpt> tag stores the latitude and longitude where the measurement was taken. The <MAC> tag stores the physical address of the AP that sent the signal. The tag <RSSI> stores the power level of that signal. To obtain the distance between the point where the data was collected and the point where the transmitting AP is located, we use the Spherical Law of Cosines. This law gives accurate results down to distances as small as 1 meter. Finally, the data was classified according to the MAC address of the identified APs and sorted from closest to farthest distance in order to obtain the observations vector $\mathbf{z} = [z_1 z_2 z_3 \cdots z_n]$.

5. Experimental Results

After data collection and model formulation phases, three blind node positions within the campus were selected to test our method. The coordinates of these positions are listed in the second column of Table 2. In order to evaluate accuracy of the results, Wi-Fi only readings and GPS only waypoints were collected at these positions. Note that this number of positions represents the best readings for both Wi-Fi and GPS signals in our experimental scenario. The overall PLE estimation can be improved as the number of blind node position readings is increased. However, this is subject to good quality readings.

Table 2

Positions	Coordinates	RMS_2	Mean HDOP
ONE	N: 5662622.36 E: 700894.86	6.2863	3.7488
TWO	N: 5662469.1 E: 700840.36	7.3534	1.5291
THREE	N: 5662503.43 E: 700915.76	3.6885	1.4245
Average		5.7760	2.2341

Before evaluating results, it is necessary to select the appropriate accuracy metrics. Since this work uses a GPS receiver to infer a power-distance model, it is necessary to employ common metrics used by GPS manufacturers such as Circular Error Probable (CEP), one-dimensional root mean square (rms_1), and two-dimensional root mean square (rms_2). See [39] for the definitions of these and other metrics.

Since the proposed localization algorithm is intended for portable electronic devices such as tablets, laptops, or handhelds, the localization takes place in a 2D map. Therefore, horizontal accuracy metric rms_2 is selected. The metric rms_2 is the square root of the average of the squared horizontal error and is calculated as follows:

$$rms_2 = \sqrt{\sigma_x^2 + \sigma_y^2}, \tag{17}$$

where σ_x^2 and σ_y^2 are the root mean square errors of the x and y components of the estimated positions, if measurements and model errors are assumed uncorrelated and the same for all observations. If this assumption is not fulfilled, the σ_x^2 and σ_y^2 are the variance of the x and y components. A related metric is Horizontal Dilution of Precision (HDOP). It is defined as the ratio of rms_2 to the root mean square of the range errors. The closer the HDOP value is to 1 the higher the accuracy obtained. These two metrics are employed to quantify errors within this section.

5.1. GPS Only Data. 7963 GPS waypoints were collected at positions ONE, 6792 at positions TWO, and lastly 8340 at position THREE. These waypoints were processed in order to extract Northing and Easting coordinates and HDOP. With this information the horizontal error position with respect to the actual coordinates of position ONE, TWO and THREE was calculated. The results for metrics rms_2 and HDOP are shown in Table 2.

5.2. Wi-Fi Only Measurements. Now, the same metrics are calculated for the WiFi only readings. In this case 5116 readings were collected at the test positions. Three landmark APs were selected for positions ONE and TWO, and only two for position THREE. Using the PLEs obtained in Section 4 for each AP, ranges to them were estimated. With the APs actual coordinates, the real ranges were calculated (listed in

Table 3

	Estimated mean ranges to APs (meters)	Real ranges to APSs (meters)	RMS_2	Mean HDOP
Position ONE	ENA_131: 43.9158 ENE_136: 60.5400 NM_110: 65.5568	38.58 68.41 91.43	15.7322	0.7866
Position TWO	KNB_125u: 48.6630 KNB_132: 51.5463 MC_184: 41.9375	51.74 75.70 41.32	32.2975	1.7112
Position THREE	MC_197: 23.2717 NM_110: 34.2559	30.09 43.91	6.0423	0.5857
Average			18.024	1.0278

Table 3). Using this information, the method derived estimated positions for the three test points. The resulting rms_2 and HDOP are summarized in Table 3.

Comparing Tables 2 and 3, it can be seen that the rms_2 measure of our results is almost 3 times the GPS rms_2 measure. It should be noted that a comparison of GPS and WiFi derived HDOP values is not appropriate as each method has its own unrelated ranging error.

6. Conclusions and Future Work

In this paper an adaptive method to formulate a power-distance model and infer distances to AP landmarks was proposed. The method is formulated as state space system. The method uses GPS and RSSI data to identify the short term and long term behavior on of the received signal power. The system's design matrix includes the ray tracing models (canyon and non-canyon) needed to estimate the PLE. Based on this the distance to each RSSI measurement source is estimated and consequently its respective position. Once the GPS-Assisted PLE model is derived at any time a user without GPS (for instance, equipped with a WiFi-only device) is capable of estimating her/his position.

A case study was implemented employing an 802.11 based network and data from a consumer-grade GPS receiver. The results were encouraging. The rms_2 error using only RSSI and the derived GPS-Assisted PLE model was, on average, 18.02 meters. On the other hand, the average error using only a consumer grade GPS was 5.77 meters. This means that the proposed method is capable of formulating propagation models that overcome signal impairments and deliver results which are approximately 3 times the GPS error. The proposal results are promising and its accuracy is higher than disable GPS Selective Availability service which consisted of 100 m horizontal.

Although there are different works in the literature review that combine GPS and RSSI, none of them explicitly produce a PLE model. However, there are related works with real implementations which obtain a higher accuracy but they require additional GPS hardware or manual deployments in predefined grids.

Future work includes additional case studies for protocols such as 802.15.4 and 802.15.1. Also, since least squares are the basis of the proposed methodology this can be extended to other applications such as blind node tracking based on Kalman Filters.

References

[1] A. Ward, A. Jones, and A. Hopper, "A new location technique for the active office," *IEEE Personal Communications*, vol. 4, no. 5, pp. 42–47, 1997.

[2] Y. C. Tseng, S. Wu, C. Chao et al., "Location awareness in ad hoc wireless mobile networks," *IEEE Computer*, vol. 34, no. 6, pp. 46–52, 2001.

[3] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *IEEE Global Telecommunicatins Conference (GLOBECOM '01)*, pp. 2926–2931, San Antonio, Tex, usa, November 2001.

[4] A. Bharathidasan and V. A. S. Ponduru, *Sensor Networks: An Overview*, Department of Computer Science, University of California, Los Angeles, Calif, USA.

[5] D. Culler, D. Estrin, and M. Srivastava, "Guest editors' introduction: overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, 2004.

[6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.

[7] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "Cricket location-support system," in *6th ACM Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 32–43, Boston, Mass, USA, August 2000.

[8] M. Kushwaha, K. Molnár, J. Sallai, P. Völgyesi, M. Maróti, and A. Lédeczi, "Sensor node localization using mobile acoustic beacons," in *2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS '05)*, pp. 483–491, November 2005.

[9] A. Savvides, H. Park, and M. B. Srivastava, "The bits and flops of the n-hop multilateration primitive for node localization problems," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA '02)*, pp. 112–121, September 2002.

[10] D. Niculescu and B. Nath, "Position and orientation in ad hoc networks," *Ad Hoc Networks*, vol. 2, no. 2, pp. 133–151, 2004.

[11] P. Kulakowski, J. Vales-Alonso, E. Egea-López, W. Ludwin, and J. García-Haro, "Angle-of-arrival localization based on antenna

arrays for wireless sensor networks," *Computers & Electrical Engineering*, vol. 36, no. 6, pp. 1181–1186, 2010.

[12] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "Active badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91–102, 1992.

[13] J. Hightower, G. Borriello, and R. Want, "SpotOn: an indoor 3d location sensing technology based on RF signal strength," Tech. Rep., IEEE Transactions on Consumer Electronics, 2000.

[14] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings of the 19th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, vol. 2, pp. 775–784, Tel Aviv, Israel, March 2000.

[15] S. Sen, R. R. Choudhury, and S. Nelakuditi, "SpinLoc: spin once to know your location," in *Proceedings of the Workshop on Mobile Computing Systems and Applications (HotMobile '12)*, San Diego Calif, USA, February 2010.

[16] S. Mazuelas, A. Bahillo, R. M. Lorenzo et al., "Robust indoor positioning provided by real-time RSSI values in unmodified WLAN networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 5, pp. 821–831, 2009.

[17] The Institute of Electrical and Electronics Engineers, *IEEE 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE-SA, Piscataway, NJ, USA, 2007.

[18] A. Neskovic, N. Neskovic, and G. Paunovic, "Modern Approaches in Modeling of Mobile Radio Systems Propagation Environment," in *IEEE Communications Surveys, Third Quarter*, IEEE Press, Piscataway, NJ, USA, 2000.

[19] E. C. L. Chan, G. Baciu, and S. C. Mak, "Using wi-fi signal strength to localize in wireless sensor networks," in *Proceedings of the WRI International Conference on Communications and Mobile Computing (CMC '09)*, pp. 538–542, January 2009.

[20] P. Prasithsangaree, P. Krishnamurthy, and P. K. Chrysanthis, "On indoor position location with wireless lans," in *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '02)*, vol. 2, pp. 720–724, September 2002.

[21] P. Castro, P. -Chiu, T. Kremeneck, and R. Muntz, "A probabilistic location service for wireless networks environments," in *Proceeding of the 3rd International Conference on Ubiquitous Computing (Ubicomp '01)*, pp. 18–24, Springer, September 2001.

[22] A. M. Ladd, K. E. Bekris, A. P. Rudys, D. S. Wallach, and L. E. Kavraki, "On the feasibility of using wireless ethernet for indoor localization," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 555–559, 2004.

[23] S. Yang, Y. Lee, R. Y. Yen et al., "A wireless LAN measurement method based on RSSI and FER," in *Proceedings of the 5th Asia-Pacific Conference on Communications and 4th Optoelectronics and Communications Conference (APCC/OECC '99)*, vol. 1, pp. 821–824, October 1999.

[24] K. Srinivasan and P. Levis, "RSSI is under appreciated," in *Proceedings of the 3rd Workshop on Embedded Networked Sensors (EmNets '06)*, May 2006.

[25] J. Latvala, J. Syrjarinne, H. Ikonen, and J. Niittylathi, "Evaluation of RSSI-based human tracking," in *Proceedings of the European Signal Processing Conference (EUPSICO '00)*, vol. 10, pp. 2273–2276, Tampere, Finlande, July 2000.

[26] J. Small, A. Smailagic, and D. Siewiorek, "Determining user location for context aware computing through the use of a wireless LAN infrastructure," Project Aura Report, http://www.ices.cmu.edu/reports/040201.pdf.

[27] P. Bergamo and G. Mazzi, "Localization in sensor networks with fading and mobility," in *Proceedings of the 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, September 2002.

[28] W. Afric, B. Zovko-Cihlar, and S. Grgic, "Methodology of path loss calculation using measurement results," in *Proceedings of the 14th International Workshop on Systems Signals and Image Processing (IWSSIP '07) and 6th EURASIP Conference Focused on Speech and Image Processing, Multimedia Communications and Services (EC-SIPMCS '07)*, pp. 257–260, June 2007.

[29] M. Bshara and L. van Biesen, "Localization in wireless networks depending on map-supported path loss model: a case study on WiMAX networks," in *Proceedings of the 2009 IEEE GLOBECOM Workshops*, pp. 1–5, December 2009.

[30] Y. S. Lu, C. F. Lai, C. C. Hu, Y. M. Huang, and X. H. Ge, "Path loss exponent estimation for indoor wireless sensor positioning," *KSII Transactions on Internet and Information Systems*, vol. 4, no. 3, pp. 243–257, 2010.

[31] R. Stoleru, T. He, and J. A. Stankovic, "Walking GPS: a practical solution for localization in manually deployed wireless sensor networks," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN '04)*, pp. 480–489, November 2004.

[32] R. Singh, M. Guainazzo, and C. S. Regazzoni, "Location determination using wlan in conjuction with GPS network (Global Positioning System)," in *Proceedings of the IEEE 59th Vehicular Technology Conference (VTC '04)*, pp. 2695–2699, May 2004.

[33] A. Ibrahim and D. Ibrahim, "Real-time GPS based outdoor WiFi localization system with map display," *Advances in Engineering Software*, vol. 41, no. 9, pp. 1080–1086, 2010.

[34] E. Hecht, *Optics*, Addison Wesley, Reading, Mass, USA, 4th edition, 2001.

[35] P. M. Shankar, *Introduction to Wireless Systems*, John Wiley and Sons, New York, NY, USA, 2001.

[36] F. P. Fontán and P. M. Espiñeira, *Modeling the Wireless Propagation Channel*, John Wiley and Sons, New York, NY, USA, 2008.

[37] W. C. Y. Lee and Y. S. Yeh, "On the estimation of the second-order statistics of log-normal fading in mobile radio environment," *IEEE Transactions on Communications*, vol. 22, no. 6, pp. 869–873, 1974.

[38] E. D. Kaplan, Ed., *Understanding GPS Principles and Applications*, Artech House, Norwood, Mass, USA, 1996.

[39] F. van Diggelen, "GNSS accuracy: lies, damn lies, and statistics," *GPS World*, vol. 18, no. 1, pp. 26–32, 2007.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 212806, 11 pages
http://dx.doi.org/10.1155/2013/212806

Research Article

Localization with Single Stationary Anchor for Mobile Node in Wireless Sensor Networks

Haipeng Qu,[1] Guojia Hou,[1] Ying Guo,[2] Ning Wang,[1] and Zhongwen Guo[1]

[1] *Department of Computer Science, Ocean University of China, Qingdao 266100, China*
[2] *School of Information Science and Technology, Qing Dao University of Science and Technology, Qingdao 266100, China*

Correspondence should be addressed to Haipeng Qu; haipeng.qu@gmail.com

Received 16 October 2012; Revised 15 January 2013; Accepted 8 February 2013

Academic Editor: Long Cheng

We proposed a localization algorithm named LSARSSI for mobile node based on RSSI (received signal strength indicator) between locating sensor node with inertia module built-in and the single anchor. Instead of directly mapping RSSI values into physical distance, contrasting RSSI values received from anchor in different visited locations, LSARSSI utilizes the geometric relationship of perpendicular intersection to compute node positions. Given that the values of RSSI among two visited locations are equal, we regard that their distances to anchor node are equal. After obtaining several sets of such visited locations, the relative location of mobile node and anchor node can be calculated. Because of the limitations of LSARSSI, we put forward an improved algorithm named ILSARSSI. Our scheme uses only one location-known anchor which is useful in low density environment without using additional hardware. The simulations show that LSARSSI achieves high accuracy and ILSARSSI performs high stability and feasibility.

1. Introduction

Wireless sensor networks (WSNs) are such popular research fields that are highly interdisciplinary and state-of-the-art [1]. It has emerged as one of the key enablers for a variety of applications such as military, environment monitoring, emergency response, target detection and tracking, and some business fields. In recent years, a number of research achievements about WSNs localization have arisen. According to the deployment of beacon nodes (which is also called anchor), the localization approaches that have been proposed can be divided into two types: one of which is based on multiple stationary beacon nodes [2–5], and another is based on mobile beacon node(s) [6–10].

But these methods are not suitable for all the applications of WSNs. In some scenarios, however, there is only a stationary beacon node [11] or a seed (reference node which position is known). For example, when we observe WSN on the sea, data can be transmitted to the shore or on board through an aggregation node, and only the position of aggregation node is known. Another example is regarding mountain inspection; the inspector would carry sensor module to move, transmitting data collected through the aggregation base station on the peak, and only the position of base station is known. In this situation, neither methods based on mobile beacon nor methods based on multiple beacon nodes could be used directly.

To address the previous issues, in this paper we proposed a localization algorithm named LSARSSI, an RSSI-based localization scheme using single stationary anchor. To increase the feasibility of our scheme, we further put forward an improved method called ILSARSSI. Major contributions of this paper are as follows.

(i) Our schemes can be used to locate mobile node in low density environment, which only need single anchor.

(ii) In order to avoid errors from directly mapping absolute RSSI values to distances, we obtain the geometrical relationship of sensors by contrasting the measured RSSI values. We then design a novel localization scheme, LSARSSI, which has a better accuracy and low overhead.

(iii) The simulation results show that our schemes perform high accuracy and feasibility, even in large-scale environment.

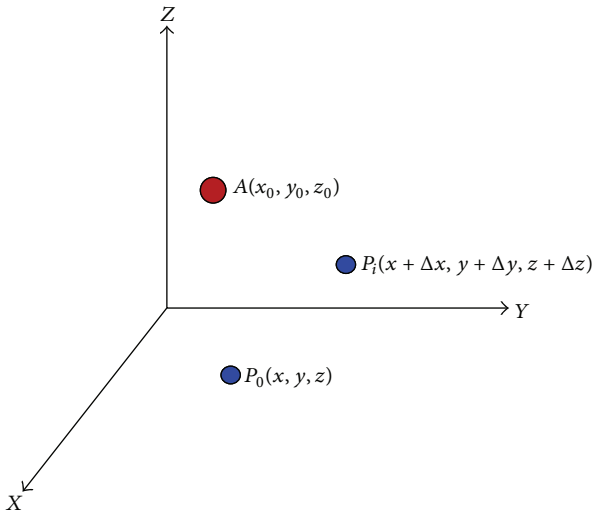


Figure 1: A model assumption in 3D.

The rest of this paper is organized as follows. Section 2 introduces the related works about the localization of WSNs. Sections 3 and 4, respectively, present our algorithm of LSARSSI and ILSARSSI. Section 5 presents our implementation and the simulation results. We conclude this paper in Section 6.

2. Related Works

According to the deployment of beacon nodes, localization technology can be classified into two categories: multiple stationary beacon nodes based approaches and mobile beacon node(s) based approaches.

2.1. Multiple Stationary Beacon Nodes Based Approaches. Measurement techniques in WSNs based on multiple stationary beacon nodes can be broadly classified into two categories: range-based approaches and range-free approaches.

2.1.1. Range-Based Approaches. Range-based approaches assume that sensor nodes can measure the distance and/or the relative directions of neighbor nodes. Several mechanisms have been proposed to measure the node's physical distance. For example, time of arrival (TOA) obtains range information through signal propagation times [12], and time difference of arrival (TDOA) estimates the node location by utilizing the time differences among signals that are received from multiple senders [13]. As an extension of TOA and TDOA, angle of arrival (AOA) allows nodes to estimate the relative directions between neighbors by setting an antenna array for each node [14].

All the previous approaches require additional hardware equipment so as to increase the cost of the sensor nodes greatly. Such, TDOA needs at least two signal generators [13]. AOA needs antenna arrays and multiple ultrasonic receivers [14].

A popular and widely used ranging technique is the received signal strength (RSS) or quantified as the received signal strength indicator (RSSI). RSSI is utilized to estimate the distance between two nodes with ordinary hardware [12, 15]. Various theoretical or empirical models of radio signal propagation have been constructed to map absolute RSSI values into estimated distances [16]. The accuracy and precision of such models, however, are far from perfect because of factors such as multipath fading and background interference [15, 17].

2.1.2. Range-Free Approaches. Given that range-based approaches are limited by hardware limitations and energy constraints, researchers have proposed range-free solutions as cost-effective alternatives.

Range-free approaches rely on the connectivity measurements between the measurement sensor nodes and a number of reference nodes, called seeds. For example, in the centroid algorithm [18], seeds broadcast their position to all neighbor nodes that record all received beacons. Each node estimates its location by calculating the center of the locations of all seeds it hears. In Approximate Point-in-Triangulation Test (APIT) [19], each node estimates whether it resides inside or outside several triangular regions bounded by the seeds that it hears and refines the computed location by overlapping such regions. In ring overlapping based on comparison of received signal strength indicator (ROCRSSI) [20], each sensor node uses a series of overlapping rings to narrow down the possible area in which it resides.

2.2. Mobile Beacons Approaches. Localization algorithms mentioned previously are in static sensor networks, which are not available in some scenarios. Recently, mobile-assisted localization approaches have been proposed to improve the efficiency of range-based approaches [10, 21]. The location of a sensor node can be calculated with the range measurements from the mobile beacon to itself; the localization accuracy can also be improved by multiple measurements that are obtained when the mobile beacons are in different positions.

Several localization schemes are proposed in this field. For example, Bergamo and Mazzini propose a scheme to perform localization, based on the estimation of the power received by only two beacons placed in known positions [22]. By starting from the received powers, eventually averaged on a given window to counteract interference and fading, the actual distance between the sensor and the beacons is derived, and the position is obtained by means of triangulation. In [10], a localization technique based on a single mobile beacon aware of its position (e.g., by being equipped with a GPS receiver) was presented. Sensor nodes receiving beacon packets infer proximity constraints to the mobile beacon and use them to construct and maintain position estimates. In PI algorithm [9], instead of using the absolute RSSI values, by contrasting the measured RSSI values from the mobile beacon to a sensor node, PI utilizes the geometric relationship of perpendicular intersection to compute the position of the node.

In recent years, more researches are focusing on mobile sensor networks, which are mainly based on mobile nodes [23–25]. Unlike other algorithms, the scene of the application

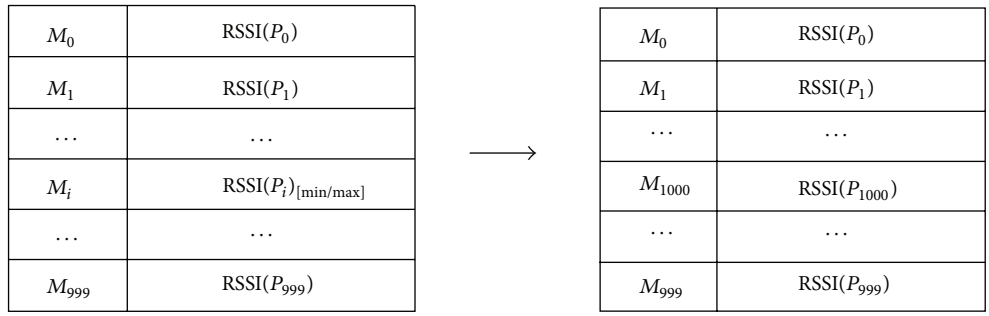


Figure 2: The 1001th message stored by sensor.

of ours is locating the mobile node with single stationary anchor by comparing the measured RSSI values between the locating sensor node with inertia module built-in and the single anchor. In this sense, LSARSSI and ILSARSSI are actually range-free approaches.

3. LSARSSI Algorithm

In this section, we first describe the application model in Section 3.1. Section 3.2 presents the design of our localization scheme in detail. Section 3.3 further presents the localization algorithm in 3D space.

3.1. Model Assumptions. For better implementing our algorithm, the hypotheses are as follows: (1) an anchor and a mobile node with inertial module whose trajectory is not designated; (2) the environment is an obstacle-free outdoor area; (3) the communication is not continuous, and the locating node receives the RSSI at different time intervals; (4) the RSSI values can be measured, and the offsets can be obtained and stored by the mobile node with inertial module.

We can illustrate it by the following model assumption. The coordinate of anchor A is (x_0, y_0, z_0). Assume that the coordinate of initial location of the locating node N is $P_0(x, y, z)$. $P_i(i \in [1, n])$ is a visited location of the mobile node N after time T, the relative offset of node N among P_0 and P_i in the X, Y, and Z axis is Δx, Δy, and Δz. So the coordinate of P_i is $(x + \Delta x, y + \Delta y, z + \Delta z)$, as shown in Figure 1.

Table 1 shows the messages obtained by the locating node N in different visited locations; it contains offsets and RSSI values. As the locating node needs to store the relative distances between locations that it visited, as well as the RSSI values between those locations to the anchor. It may require large memory storage for sensors. The issue can be resolved by the following method: assume that sensor can store 1000 messages of P_0 to P_{999}, and if the RSSI value of P_i is minimum (or maximum), the message of P_{1000} will replace it as shown in Figure 2.

As the communication is not continuous, and the visited locations for our schemes needed are countable, the computation and communication costs are acceptable.

Table 1: Messages stored by sensor.

Locations	Messages (node N)			
P_0	Δx	Δy	Δz	RSSI(P_0)
P_1	$x + \Delta x_{01}$	$y + \Delta y_{01}$	$z + \Delta z_{01}$	RSSI(P_1)
⋮	⋮	⋮	⋮	⋮
P_i	$x + \Delta x_{0i}$	$y + \Delta y_{0i}$	$z + \Delta z_{0i}$	RSSI(P_i)
⋮	⋮	⋮	⋮	⋮
P_{n-1}	$x + \Delta x_{0n-1}$	$y + \Delta y_{0n-1}$	$z + \Delta z_{0n-1}$	RSSI(P_{n-1})
P_n	$x + \Delta x_{0n}$	$y + \Delta y_{0n}$	$z + \Delta z_{0n}$	RSSI(P_n)

3.2. Localization Algorithm Design. Typically, the ensemble mean received power in a real world obstructed channel decays proportional to d^{-np}, where n_p is the path-loss exponent [26]. The ensemble mean power at distance d is typically modeled as

$$\left[\overline{p_r(d)}\right]_{dBm} = \left[p_r(d_0)\right]_{dBm} - 10n \log \frac{d}{d_0}, \quad (1)$$

where P_0 is the received power (dBm) at short reference distance d_0.

From (1) we know that, ideally, the longer the distance between sender and receiver, the weaker the signal strength detected by the receiver. The localization scheme was inspired by the *perpendicular bisector of a chord conjecture* [27]. With two chords of the same circle, the intersection point of two perpendicular bisectors of the chords will be the center of the circle. The localization problem can be transformed based on the conjecture. The center of the circle is the position of the anchor; the chord is a segment that is a connection of two positions (the RSSI values of the two positions are equal) of the mobile node at a given moment. The solution is detailed illustrated in Figure 3.

Node A is the anchor, N is the mobile locating node, P_0, P_1, P_2, and P_3 are the visited locations of locating node N after the time period of it, jt, kt, lt; B, C are the midpoints of segments P_0P_1, P_2P_3, and RSSI(P_0) = RSSI(P_1), RSSI(P_2) = RSSI(P_3).

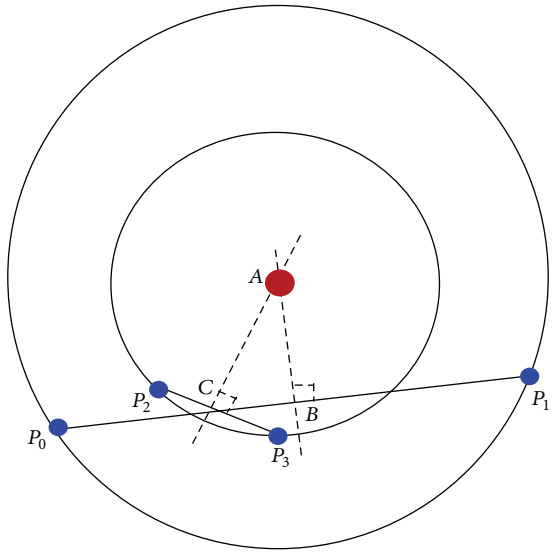


Figure 3: An example of LSARSSI algorithm in 2D.

The following are the formulas for calculating the location of locating node:

$$\begin{aligned} A &= (x_0, y_0), \qquad P_0 = (x, y), \\ P_1 &= (x + \Delta x_{01}, y + \Delta y_{01}), \\ P_2 &= (x + \Delta x_{02}, y + \Delta y_{02}), \\ P_3 &= (x + \Delta x_{03}, y + \Delta y_{03}), \\ B &= (x + \Delta x_{0B}, y + \Delta y_{0B}), \\ C &= (x + \Delta x_{0C}, y + \Delta y_{0C}). \end{aligned} \tag{2}$$

According to the geometry of vector, we can obtain that

$$\begin{aligned} &\overrightarrow{P_0P_1} = (\Delta x_{01}, \Delta y_{01}), \qquad \overrightarrow{P_2P_3} = (\Delta x_{23}, \Delta y_{23}), \\ &\overrightarrow{AB} = (x + \Delta x_{0B} - x_0, y + \Delta y_{0B} - y_0), \\ &\overrightarrow{AC} = (x + \Delta x_{0C} - x_0, y + \Delta y_{0C} - y_0), \\ &\overrightarrow{P_0P_1} \cdot \overrightarrow{AB} = \Delta x_{01} * (x + \Delta x_{0B} - x_0) \\ &\qquad + \Delta y_{01} * (y + \Delta y_{0B} - y_0) = 0, \\ &\overrightarrow{P_2P_3} \cdot \overrightarrow{AC} = \Delta x_{23} * (x + \Delta x_{0C} - x_0) \\ &\qquad + \Delta y_{23} * (y + \Delta y_{0C} - y_0) = 0, \\ &\qquad \Downarrow \\ &(1)\ \Delta x_{01} * x + \Delta y_{01} * y \\ &\qquad = \Delta x_{01} * (x_0 - \Delta x_{0B}) + \Delta y_{01} * (y_0 - \Delta y_{0B}), \end{aligned}$$

$$\begin{aligned} &(2)\ \Delta x_{23} * x + \Delta y_{23} * y \\ &\qquad = \Delta x_{23} * (x_0 - \Delta x_{0C}) + \Delta y_{23} * (y_0 - \Delta y_{0C}). \end{aligned} \tag{3}$$

Assume that

$$\begin{aligned} a &= \Delta x_{01} * (x_0 - \Delta x_{0B}) \\ &\quad + \Delta y_{01} * (y_0 - \Delta y_{0B}), \\ b &= \Delta x_{23} * (x_0 - \Delta x_{0C}) \\ &\quad + \Delta y_{23} * (y_0 - \Delta y_{0C}). \end{aligned} \tag{4}$$

then (1), (2) can be converted into

$$\begin{aligned} &(3)\ \Delta x_{01} * x + \Delta y_{01} * y = a, \\ &(4)\ \Delta x_{23} * x + \Delta y_{23} * y = b, \\ &\qquad \Downarrow \\ &x = \frac{D_1}{D}, \qquad y = \frac{D_2}{D}, \end{aligned} \tag{5}$$

where

$$\begin{aligned} D &= \begin{vmatrix} \Delta x_{01} & \Delta y_{01} \\ \Delta x_{23} & \Delta y_{23} \end{vmatrix}, \qquad D_1 = \begin{vmatrix} a & \Delta y_{01} \\ b & \Delta y_{23} \end{vmatrix}, \\ D_2 &= \begin{vmatrix} \Delta x_{01} & a \\ \Delta x_{23} & b \end{vmatrix}. \end{aligned} \tag{6}$$

3.3. Algorithm in 3D. Here we extend our algorithm to three-dimensional space. As we know, if there are three noncoincident chords on the sphere, the intersection of three midvertical planes is the center of the sphere provided that these three planes can intersect.

The general idea of this algorithm in 3D utilizes the previous conjecture. This problem can be described as follows: the locations of locating node N at different times compose a collection $P = \{P_0, P_1 \dots P_n\}$. Their RSSI values are $\{\mathrm{RSSI}(P_0), \mathrm{RSSI}(P_1) \dots \mathrm{RSSI}(P_n)\}$, locations can be divided into a number of subsets $\{N_1, N_2 \dots N_m\}$ based on RSSI values, and the RSSI values in each subset are equal, respectively. After that, given that one of the following conditions occurs, our LSARSSI algorithm can be achieved: (1) $|N_a| = 2$, and $|N_b| = 2$ and $|N_c| = 2$ (as shown in Figure 4(a)); (2) $|N_a| = 3$ and $|N_b| = 2$ (as shown in Figure 4(b)); (3) $|N_a| = 4$ (as shown in Figure 4(c)) $(a, b, c \in [1, m])$.

As (2) and (3) are the special cases of (1), here we take $|N_a| = 2$, $|N_b| = 2$ and $|N_c| = 2$ as an example to indicate formulas for calculating the location of locating node. A is the anchor; P_0, P_1, P_2, P_3, P_4, and P_5 are the visited locations of the mobile node; B, C, and D are the midpoint of P_0P_1, P_2P_3, and P_4P_5. And

$$\begin{aligned} A &= (x_0, y_0, z_0), \\ P_0 &= (x, y, z), \\ P_1 &= (x + \Delta x_{01}, y + \Delta y_{01}, z + \Delta z_{01}), \\ P_2 &= (x + \Delta x_{02}, y + \Delta y_{02}, z + \Delta z_{02}), \end{aligned}$$

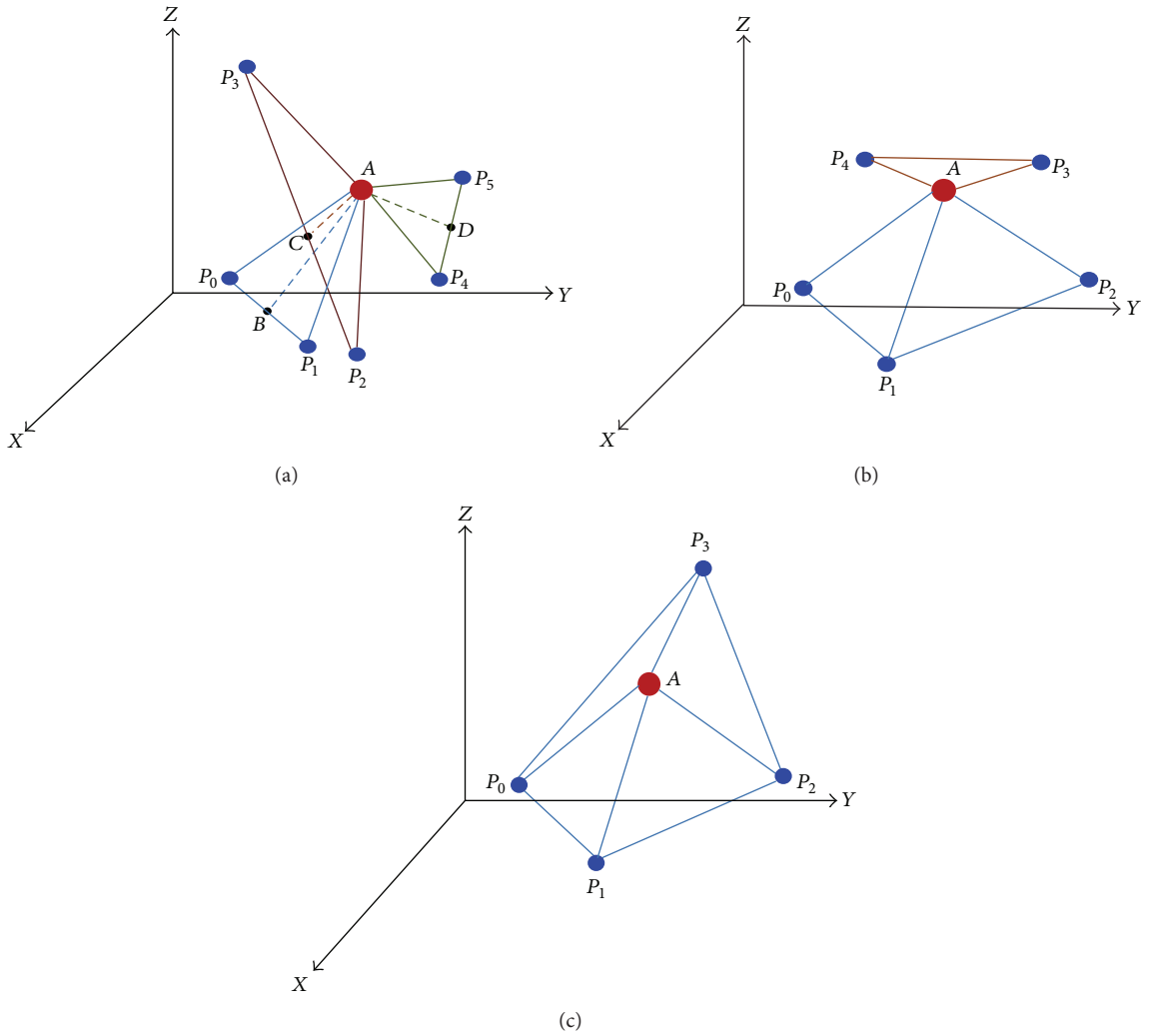


Figure 4: Examples of LSARSSI algorithm in 3D.

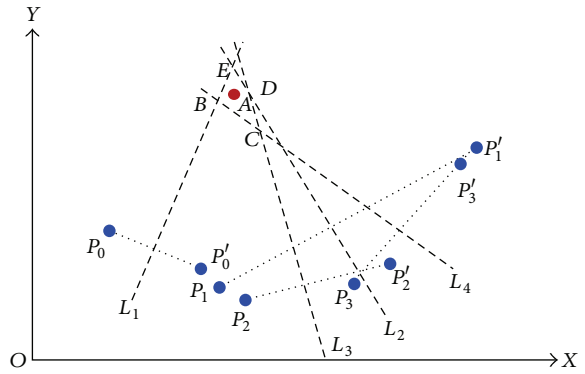


Figure 5: An example of ILSARSSI algorithm in 2D.

$$P_3 = (x + \Delta x_{03}, y + \Delta y_{03}, z + \Delta z_{03}),$$
$$P_4 = (x + \Delta x_{04}, y + \Delta y_{04}, z + \Delta z_{04}),$$
$$P_5 = (x + \Delta x_{05}, y + \Delta y_{05}, z + \Delta z_{05}),$$

$$\begin{aligned} \mathrm{RSSI}(P_0) &= \mathrm{RSSI}(P_1), \\ \mathrm{RSSI}(P_2) &= \mathrm{RSSI}(P_3), \\ \mathrm{RSSI}(P_4) &= \mathrm{RSSI}(P_5). \end{aligned} \tag{7}$$

According to the geometry of vector, we can obtain that

$$\overrightarrow{P_0P_1} = (\Delta x_{01}, \Delta y_{01}, \Delta z_{01}), \qquad \overrightarrow{P_2P_3} = (\Delta x_{23}, \Delta y_{23}, \Delta z_{23}),$$
$$\overrightarrow{P_4P_5} = (\Delta x_{45}, \Delta y_{45}, \Delta z_{45}),$$
$$\overrightarrow{AB} = (x + \Delta x_{0B} - x_0, y + \Delta y_{0B} - y_0, z + \Delta z_{0B} - z_0),$$
$$\overrightarrow{AC} = (x + \Delta x_{0C} - x_0, y + \Delta y_{0C} - y_0, z + \Delta z_{0C} - z_0),$$
$$\Downarrow$$
$$\overrightarrow{AD} = (x + \Delta x_{0D} - x_0, y + \Delta y_{0D} - y_0, z + \Delta z_{0D} - z_0),$$

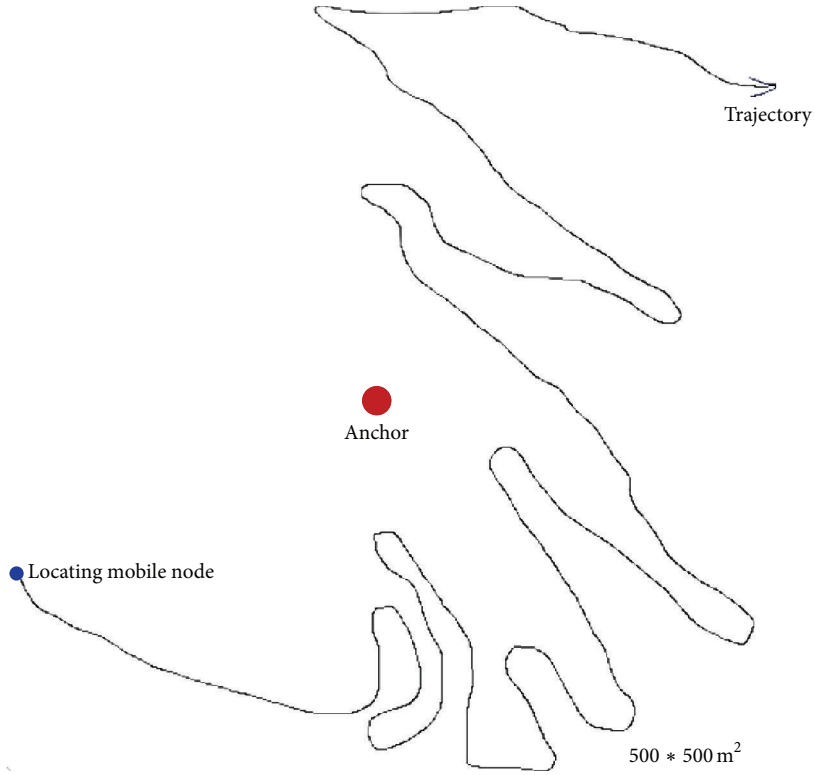


Figure 6: A trajectory of the locating mobile node.

$$\overrightarrow{P_0P_1} \cdot \overrightarrow{AB} = \Delta x_{01} * (x + \Delta x_{0B} - x_0) + \Delta y_{01} * (y + \Delta y_{0B} - y_0) + \Delta z_{01} * (z + \Delta z_{0B} - z_0) = 0,$$

$$\overrightarrow{P_2P_3} \cdot \overrightarrow{AC} = \Delta x_{23} * (x + \Delta x_{0C} - x_0) + \Delta y_{23} * (y + \Delta y_{0C} - y_0) + \Delta z_{23} * (z + \Delta z_{0C} - z_0) = 0,$$

$$\Downarrow$$

$$\overrightarrow{P_4P_5} \cdot \overrightarrow{AD} = \Delta x_{45} * (x + \Delta x_{0D} - x_0) + \Delta y_{45} * (y + \Delta y_{0D} - y_0) + \Delta z_{45} * (z + \Delta z_{0D} - z_0) = 0,$$

$$(1)\ \Delta x_{01} * x + \Delta y_{01} * y + \Delta z_{01} * z = \Delta x_{01} * (x_0 - \Delta x_{0B}) + \Delta y_{01} * (y_0 - \Delta y_{0B}) + \Delta z_{01} * (z_0 - \Delta z_{0B}),$$

$$(2)\ \Delta x_{23} * x + \Delta y_{23} * y + \Delta z_{23} * z = \Delta x_{23} * (x_0 - \Delta x_{0C}) + \Delta y_{23} * (y_0 - \Delta y_{0C}) + \Delta z_{23} * (z_0 - \Delta z_{0C}),$$

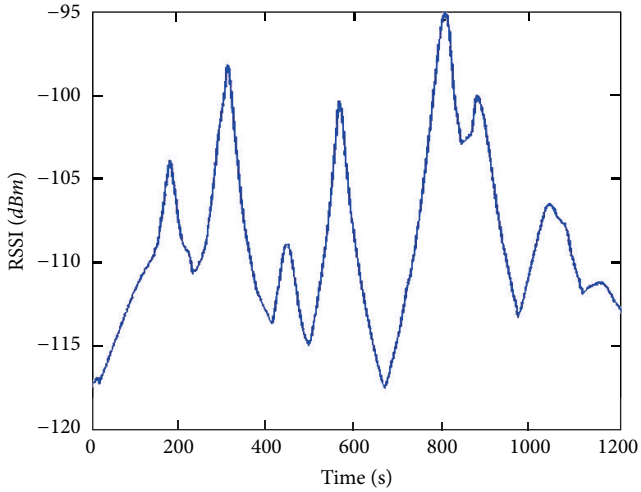
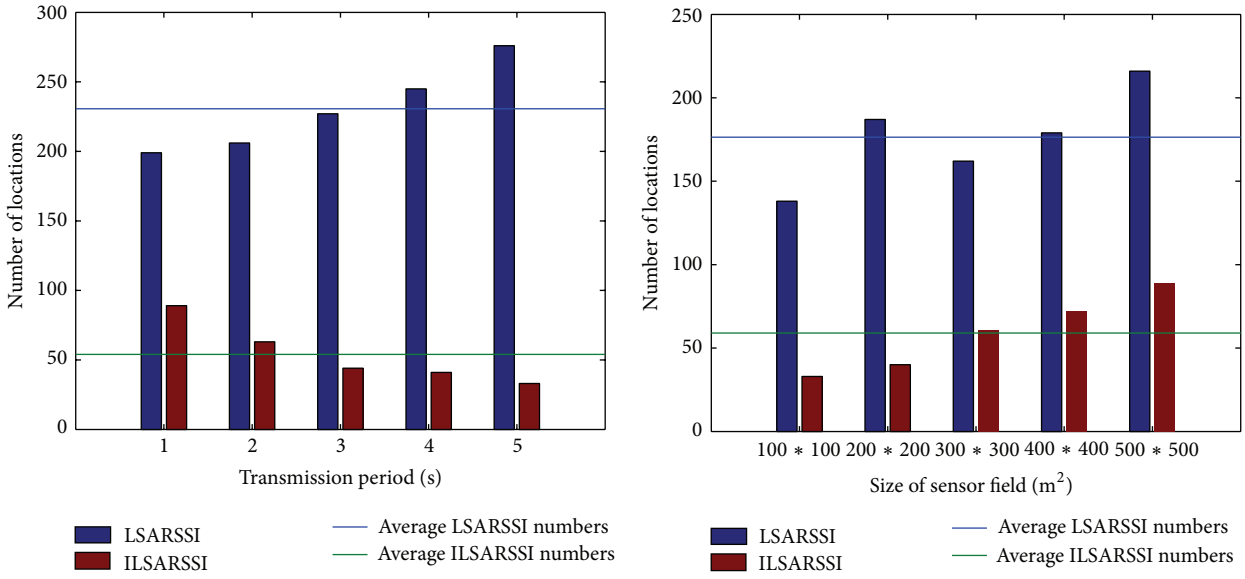


Figure 7: RSSI values received by locating node after time T.

$$(3)\ \Delta x_{45} * x + \Delta y_{45} * y + \Delta z_{45} * z = \Delta x_{45} * (x_0 - \Delta x_{0D}) + \Delta y_{45} * (y_0 - \Delta y_{0D}) + \Delta z_{45} * (z_0 - \Delta z_{0D}). \quad (8)$$

Assume that

$$a = \Delta x_{01} * (x_0 - \Delta x_{0B}) + \Delta y_{01} * (y_0 - \Delta y_{0B}) + \Delta z_{01} * (z_0 - \Delta z_{0B}),$$



(a) Impact of packet transmission period

(b) Impact of size of sensor field

Figure 8: Number of locations of LSARSSI and ILSARSSI algorithms.

$$
\begin{aligned}
b &= \Delta x_{23} * (x_0 - \Delta x_{0C}) + \Delta y_{23} * (y_0 - \Delta y_{0C}) \\
&\quad + \Delta z_{23} * (z_0 - \Delta z_{0C}), \\
c &= \Delta x_{45} * (x_0 - \Delta x_{0D}) + \Delta y_{45} * (y_0 - \Delta y_{0D}) \\
&\quad + \Delta z_{45} * (z_0 - \Delta z_{0D}),
\end{aligned} \tag{9}
$$

then (1), (2), and (3) can be converted into

$$
\begin{aligned}
&(4)\ \Delta x_{01} * x + \Delta y_{01} * y + \Delta z_{01} * z = a, \\
&(5)\ \Delta x_{23} * x + \Delta y_{23} * y + \Delta z_{23} * z = b, \\
&(6)\ \Delta x_{45} * x + \Delta y_{45} * y + \Delta z_{45} * z = c, \\
&\Downarrow \\
&x = \frac{D_1}{D}, \qquad y = \frac{D_2}{D}, \qquad z = \frac{D_3}{D},
\end{aligned} \tag{10}
$$

where

$$
\begin{aligned}
D &= \begin{vmatrix} \Delta x_{01} & \Delta y_{01} & \Delta z_{01} \\ \Delta x_{23} & \Delta y_{23} & \Delta z_{23} \\ \Delta x_{45} & \Delta y_{45} & \Delta z_{45} \end{vmatrix}, \qquad D_1 = \begin{vmatrix} a & \Delta y_{01} & \Delta z_{01} \\ b & \Delta y_{23} & \Delta z_{23} \\ c & \Delta y_{45} & \Delta z_{45} \end{vmatrix}, \\
D_2 &= \begin{vmatrix} \Delta x_{01} & a & \Delta z_{01} \\ \Delta x_{23} & b & \Delta z_{23} \\ \Delta x_{45} & c & \Delta z_{45} \end{vmatrix}, \qquad D_3 = \begin{vmatrix} \Delta x_{01} & \Delta y_{01} & a \\ \Delta x_{23} & \Delta y_{23} & b \\ \Delta x_{45} & \Delta y_{45} & c \end{vmatrix}.
\end{aligned} \tag{11}
$$

4. Improved Algorithm: ILSARSSI

In real scenarios, the measure of the RSS is discontinuous; it may be difficult to obtain several groups of locations with the same RSSI values. So the algorithm proposed previously cannot be successfully used to locate. However, the locating node can be located by comparing the RSSI according to the *signal attenuation model* [26].

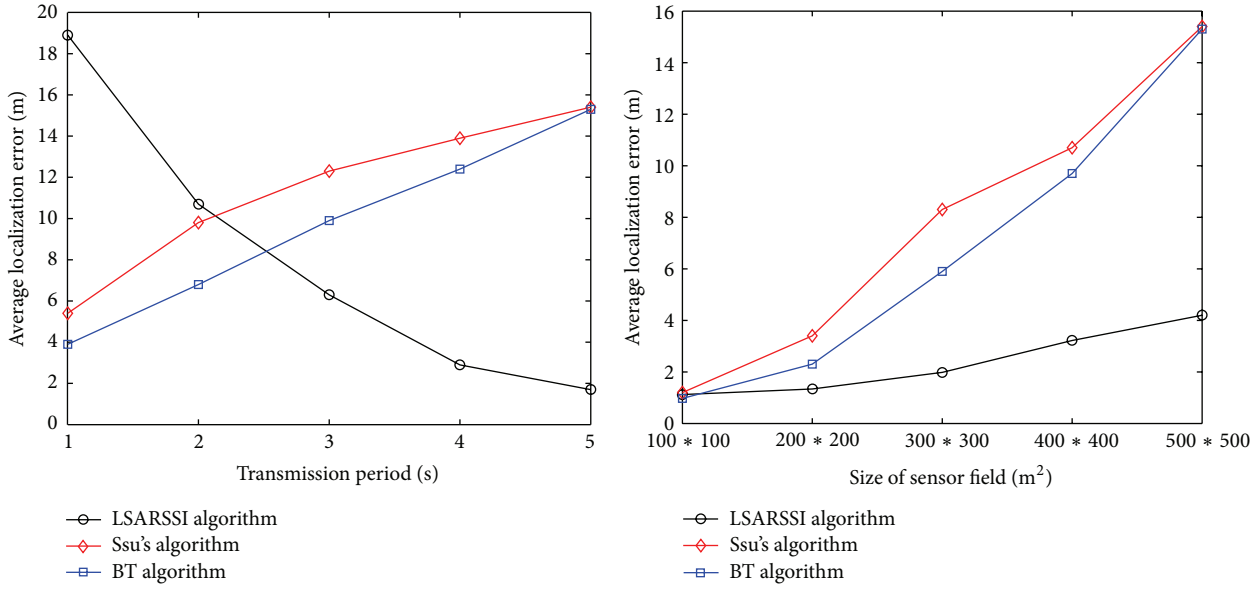
Figure 5 demonstrates an example of the improved algorithm ILSARSSI. The RSSI values of node N measured in P_0, P_0', P_1, P_1', P_2, P_2', P_3, P_3' are RSSI(P_0), RSSI(P_0'), RSSI(P_1), RSSI(P_1'), RSSI(P_2), RSSI(P_2'), RSSI(P_3), RSSI(P_3'), and RSSI(P_0) < RSSI(P_0'), RSSI(P_1) > RSSI(P_1'), RSSI(P_2) > RSSI(P_2'), RSSI(P_3) > RSSI(P_3'). The anchor is in the enclosed region BCDE of four perpendicular bisectors of the segment P_0P_0', P_1P_1', P_2P_2', P_3P_3'. After finding a sufficient number of locations that meet such condictions, we can further narrow the area where the anchor node is. We take the centroid of the enclosed area as the position of the anchor, and then the coordinates of visited locations can be obtained.

The following are the formulas for calculating the location of locating node:

$$
\begin{aligned}
&A(x_0, y_0), \qquad P_0 = (x_1, y_1), \qquad P_0' = (x_1', y_1'), \\
&\qquad\qquad P_1 = (x_2, y_2), \qquad P_1' = (x_2', y_2'), \\
&\qquad\qquad P_2 = (x_3, y_3), \qquad P_2' = (x_3', y_3'), \\
&\qquad\qquad P_3 = (x_4, y_4), \qquad P_3' = (x_4', y_4').
\end{aligned} \tag{12}
$$

As L_1, L_2, L_3, and L_4 are the perpendicular bisector of P_0P_0', P_1P_1', P_2P_2', and P_3P_3', according to the geometry of vector, we can obtain that

$$
\begin{aligned}
L_1 &: (x_1 - x_1') * x + (y_1 - y_1') * y \\
&= \left(\frac{x_1 + x_1'}{2}\right) * (x_1 - x_1') + \left(\frac{y_1 + y_1'}{2}\right) * (y_1 - y_1'),
\end{aligned}
$$



(a) Impact of packet transmission period

(b) Impact of size of sensor field

FIGURE 9: Localization error of our LSARSSI algorithm.

$$
\begin{aligned}
L_2 &: \left(x_2 - x_2'\right) * x + \left(y_2 - y_2'\right) * y \\
&= \left(\frac{x_2 + x_2'}{2}\right) * \left(x_2 - x_2'\right) + \left(\frac{y_2 + y_2'}{2}\right) * \left(y_2 - y_2'\right), \\
L_3 &: \left(x_3 - x_3'\right) * x + \left(y_3 - y_3'\right) * y \\
&= \left(\frac{x_3 + x_3'}{2}\right) * \left(x_3 - x_3'\right) + \left(\frac{y_3 + y_3'}{2}\right) * \left(y_3 - y_3'\right), \\
L_4 &: \left(x_4 - x_4'\right) * x + \left(y_4 - y_4'\right) * y \\
&= \left(\frac{x_4 + x_4'}{2}\right) * \left(x_4 - x_4'\right) + \left(\frac{y_4 + y_4'}{2}\right) * \left(y_4 - y_4'\right).
\end{aligned}
$$

As $L_1 \cap L_4 = B$, $L_3 \cap L_4 = C$,

$L_2 \cap L_3 = D$, $L_1 \cap L_2 = E$,

$$\Downarrow \tag{13}$$

$$
x_B = \frac{\begin{vmatrix} \left(\frac{x_1 + x_1'}{2}\right) * \left(x_1 - x_1'\right) + \left(\frac{y_1 + y_1'}{2}\right) * \left(y_1 - y_1'\right) & y_1 - y_1' \\ \left(\frac{x_4 + x_4'}{2}\right) * \left(x_4 - x_4'\right) + \left(\frac{y_4 + y_4'}{2}\right) * \left(y_4 - y_4'\right) & y_4 - y_4' \end{vmatrix}}{\begin{vmatrix} x_1 - x_1' & y_1 - y_1' \\ x_4 - x_4' & y_4 - y_4' \end{vmatrix}},
$$

$$
x_C = \frac{\begin{vmatrix} \left(\frac{x_2 + x_2'}{2}\right) * \left(x_2 - x_2'\right) + \left(\frac{y_2 + y_2'}{2}\right) * \left(y_2 - y_2'\right) & y_2 - y_2' \\ \left(\frac{x_4 + x_4'}{2}\right) * \left(x_4 - x_4'\right) + \left(\frac{y_4 + y_4'}{2}\right) * \left(y_4 - y_4'\right) & y_4 - y_4' \end{vmatrix}}{\begin{vmatrix} x_2 - x_2' & y_2 - y_2' \\ x_4 - x_4' & y_4 - y_4' \end{vmatrix}},
$$

$$
x_D = \frac{\begin{vmatrix} \left(\frac{x_2 + x_2'}{2}\right) * \left(x_2 - x_2'\right) + \left(\frac{y_2 + y_2'}{2}\right) * \left(y_2 - y_2'\right) & y_2 - y_2' \\ \left(\frac{x_3 + x_3'}{2}\right) * \left(x_3 - x_3'\right) + \left(\frac{y_3 + y_3'}{2}\right) * \left(y_3 - y_3'\right) & y_3 - y_3' \end{vmatrix}}{\begin{vmatrix} x_2 - x_2' & y_2 - y_2' \\ x_3 - x_3' & y_3 - y_3' \end{vmatrix}},
$$

$$
x_E = \frac{\begin{vmatrix} \left(\frac{x_1 + x_1'}{2}\right) * \left(x_1 - x_1'\right) + \left(\frac{y_1 + y_1'}{2}\right) * \left(y_1 - y_1'\right) & y_1 - y_1' \\ \left(\frac{x_2 + x_2'}{2}\right) * \left(x_2 - x_2'\right) + \left(\frac{y_2 + y_2'}{2}\right) * \left(y_2 - y_2'\right) & y_2 - y_2' \end{vmatrix}}{\begin{vmatrix} x_1 - x_1' & y_1 - y_1' \\ x_2 - x_2' & y_2 - y_2' \end{vmatrix}},
$$

$$
y_B = \frac{\begin{vmatrix} x_1 - x_1' & \left(\frac{x_1 + x_1'}{2}\right) * \left(x_1 - x_1'\right) + \left(\frac{y_1 + y_1'}{2}\right) * \left(y_1 - y_1'\right) \\ x_4 - x_4' & \left(\frac{x_4 + x_4'}{2}\right) * \left(x_4 - x_4'\right) + \left(\frac{y_4 + y_4'}{2}\right) * \left(y_4 - y_4'\right) \end{vmatrix}}{\begin{vmatrix} x_1 - x_1' & y_1 - y_1' \\ x_4 - x_4' & y_4 - y_4' \end{vmatrix}},
$$

$$y_C = \frac{\begin{vmatrix} x_2 - x_2' \left(\frac{x_2 + x_2'}{2}\right) * (x_2 - x_2') + \left(\frac{y_2 + y_2'}{2}\right) * (y_2 - y_2') \\ x_4 - x_4' \left(\frac{x_4 + x_4'}{2}\right) * (x_4 - x_4') + \left(\frac{y_4 + y_4'}{2}\right) * (y_4 - y_4') \end{vmatrix}}{\begin{vmatrix} x_2 - x_2' & y_2 - y_2' \\ x_4 - x_4' & y_4 - y_4' \end{vmatrix}},$$

$$y_D = \frac{\begin{vmatrix} x_2 - x_2' \left(\frac{x_2 + x_2'}{2}\right) * (x_2 - x_2') + \left(\frac{y_2 + y_2'}{2}\right) * (y_2 - y_2') \\ x_3 - x_3' \left(\frac{x_3 + x_3'}{2}\right) * (x_3 - x_3') + \left(\frac{y_3 + y_3'}{2}\right) * (y_3 - y_3') \end{vmatrix}}{\begin{vmatrix} x_2 - x_2' & y_2 - y_2' \\ x_3 - x_3' & y_3 - y_3' \end{vmatrix}},$$

$$y_E = \frac{\begin{vmatrix} x_1 - x_1' \left(\frac{x_1 + x_1'}{2}\right) * (x_1 - x_1') + \left(\frac{y_1 + y_1'}{2}\right) * (y_1 - y_1') \\ x_2 - x_2' \left(\frac{x_2 + x_2'}{2}\right) * (x_2 - x_2') + \left(\frac{y_2 + y_2'}{2}\right) * (y_2 - y_2') \end{vmatrix}}{\begin{vmatrix} x_1 - x_1' & y_1 - y_1' \\ x_2 - x_2' & y_2 - y_2' \end{vmatrix}},$$

$$\Downarrow$$

$$x_0 = \frac{x_B + x_c + x_D + x_E}{4},$$

$$y_0 = \frac{y_B + y_c + y_D + y_E}{4}. \tag{14}$$

The approach can also be applied to the localization of mobile node in three-dimensional space. The simulation results show that ILSARSSI performs high feasibility and practicality.

5. Simulation Results

To evaluate the performance of our proposed approaches, we use MATLAB 7.0 to conduct the simulations. In the following, the simulation parameters are listed in Table 2. We assume that the trajectory of the mobile locating node is random (Figure 6 is a trajectory of the locating node), the packet transmission period among anchor and mobile locating node is in a certain time interval (1 s, 2 s, 3 s, 4 s, and 5 s), the average moving speed of the locating node is 2 m/s, and the size of sensor field is in an obstacle-free area of $100 * 100\,m^2$, $200 * 200\,m^2$, $300 * 300\,m^2$, $400 * 400\,m^2$, and $500 * 500\,m^2$. Figure 7 plots the RSSI values received by locating node along the trajectory shown in Figure 6.

In the first section, we mainly validate the feasibility of LSARSSI and LSARSSI algorithm, and in the second section, we discuss the localization error of LSARSSI and ILSARSSI algorithms.

Table 2: Simulation parameters.

Parameters	Value(s)
Packet transmission period (s)	1, 2, 3, 4, 5
Moving speed (m/s)	2
Size of sensor field (m^2)	$100 * 100$, $200 * 200$, $300 * 300$, $400 * 400$, $500 * 500$

5.1. Number of Locations Simulations. Figure 8 compares the number of locations of LSARSSI algorithm and ILSARSSI algorithm. The average number of locations of ILSARSSI is much fewer than LSARSSI algorithm, which demonstrates that the ILSARSSI algorithm performs high feasibility.

(1) *Number of Locations versus Packet Transmission Period.* The packet transmission period is the time interval of transmitting message among the anchor and locating node. Figure 8(a) indicates that the number of locations of ILSARSSI algorithm decreases with increasing the packet transmission period. The average numbers of locations of ILSARSSI algorithm and ILSARSSI algorithm are approximately 240 and 55 at different transmission period, respectively.

(2) *Number of Locations versus Size of Sensor Field.* The size of sensor field is the moving range of the locating node. Figure 8(b) shows that the number of locations of ILSARSSI algorithm increases with expanding the size of sensor field.

5.2. Average Localization Error Simulations

(1) *Average Localization Error versus Packet Transmission Period.* Figure 9 compares average localization error for Ssu's [28], BT [29], and our algorithm. Figure 9(a) shows that the localization accuracy for both Ssu's and BT algorithms can be improved by reducing the packet transmission period. However, our LSARSSI algorithm performs higher accuracy with increasing packet transmission period, which can reduce the communication cost and energy consumption. If packet transmission period is 5 s, the average localization error of Ssu's and BT is approximately 15 m, but our algorithm is less than 3 m.

(2) *Average Localization Error versus Size of Sensor Field.* Figure 9(b) shows the impact of moving range on the localization error, as the size of sensor field expands from $100 * 100\,m^2$ to $500 * 500\,m^2$. The increased size of sensor field makes the Ssu's and BT algorithms less accurate, but our localization error remains within 5 m.

(3) *Average Localization Error of LSARSSI and ILSARSSI.* Figure 10 demonstrates the localization error of ILSARSSI algorithm depends on the locating time. With increasing the locating time, the locating accuracy becomes more and more precise. That is because the possible location falls into a smaller enclosed region. However, the estimated error of LSARSSI algorithm varies slightly with the increase of locating time.

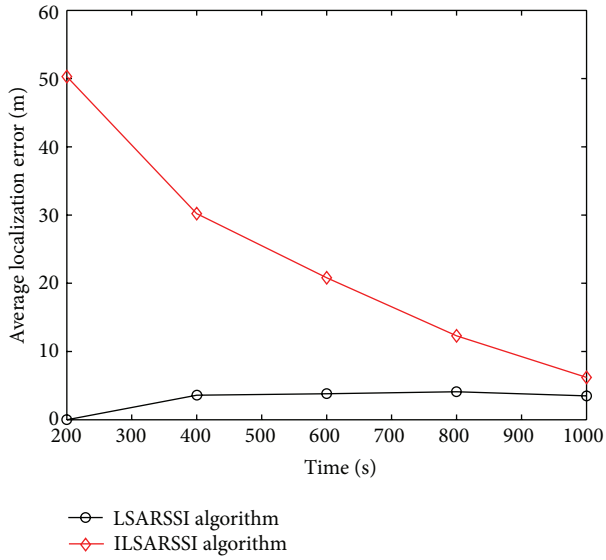


FIGURE 10: Average localization error of LSARSSI and ILSARSSI.

6. Conclusions

In this paper, we propose a localization algorithm named LSARSSI for mobile node with single anchor by aid of inertia module. The simulation results demonstrate that our LSARSSI algorithm outperforms than other range-free localization mechanisms, for example, Ssu's and BT algorithms. As the number of locations needed is approximately 200, we further proposed an improved algorithm named ILSARSSI. ILSARSSI utilize the *signal attenuation model to* narrow the region of the anchor, and finally we regard the centroid of the enclosed region as its physical position. The average number of locations of ILSARSSI algorithm is much fewer which performs high feasibility. Our scheme uses only one location-known anchor which is useful in low density environment without using additional hardware. Because our algorithms are adapted to the obstacle-free outdoor scenario, our future work will focus on indoor environment.

Acknowledgment

This work was supported by the National Natural Science Funds (Grant no. 60970129), the Public Science and Technology Research Funds Project of Ocean (Grant no. 201105033), and the Localization Algorithms of Underwater Sensor Networks Research Funds (Grant no. 61170258).

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] A. Youssef, A. Agrawala, and M. Younis, "Accurate anchor-free node localization in wireless sensor networks," in *24th IEEE International Performance, Computing, and Communications Conference (IPCCC '05)*, pp. 465–470, April 2005.

[3] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '99)*, pp. 1322–1328, Detroit, Mich, USA, May 1999.

[4] N. Bulusu, D. Estrin, L. Girod, and J. Heidemann, "Scalable coordination for wireless sensor networks: self-configuring localization systems," in *International Symposium on Communication Theory and Applications (ISCTA '01)*, Ambleside, UK, July 2001.

[5] N. Bulusu, J. Heidemann, D. Estrin, and T. Tran, "Self-configuring localization systems: design and experimental evalution," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 3, pp. 24–60, 2004.

[6] A. Iqbal, *Increasing Localization Precision in Sensor Networks with Mobile Beacons—A Genetic path planning Approach*, Research Commons Library, 2009.

[7] T. V. Srinath, A. K. Katti, and V. S. Ananthanarayna, "Analysis of mobile beacon aided in-range localization scheme in ad hoc wireless sensor networks," in *International Wireless Communications and Mobile Computing Conference (IWCMC '06)*, pp. 1159–1164, July 2006.

[8] K. Kim and W. Lee, "MBAL: a mobile beacon-assisted localization scheme for wireless sensor networks," in *16th International Conference on Computer Communications and Networks (ICCCN '07)*, pp. 57–62, August 2007.

[9] G. Zhongwen, G. Ying, H. Feng et al., "Perpendicular intersection: locating wireless sensors with mobile beacon," in *Real-Time Systems Symposium (RTSS '08)*, pp. 93–102, December 2008.

[10] M. L. Sichitiu and V. Ramadurai, "Localization of wireless sensor networks with a mobile beacon," in *IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, pp. 174–183, Fort Lauderdale, Fla, USA, October 2004.

[11] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proceedings of the Tenth Annual International Conference on Mobile Computing and Networking (MobiCom '04)*, pp. 45–57, New York, NY, USA, October 2004.

[12] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *19th IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '00)*, pp. 775–784, Tel Aviv, Israel, March 2000.

[13] A. Savvides, C. C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *7th Annual International Conference on Mobile Computing and Networking*, pp. 166–179, New York, NY, USA, July 2001.

[14] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," in *22nd IEEE Annual Joint Conference on the Computer and Communications Societies(INFOCOM '03)*, pp. 1734–1743, April 2003.

[15] J. Hightower, R. Want, and G. Borriello, *SpotON: An Indoor 3-D Location Sensing Technology Based on RF Signal Strength*, University of Washington, Seattle, Wash, USA, 2000.

[16] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1999.

[17] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Models and solutions for radio irregularity in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 2, pp. 221–262, 2006.

[18] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, 2000.

[19] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking (MobiCom '03)*, pp. 81–95, New York, NY, USA, September 2003.

[20] C. Liu, K. Wu, and T. He, "Sensor localization with ring overlapping based on comparison of received signal strength indicator," in *IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*, pp. 516–518, October 2004.

[21] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. Teller, "Mobile-assisted localization in wireless sensor networks," in *IEEE Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, pp. 172–183, March 2005.

[22] P. Bergamo and G. Mazzini, "Localization in sensor networks with fading and mobility," in *13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '02)*, pp. 750–754, September 2002.

[23] L. Mihaylova, D. Angelova, D. Bull, and N. Canagarajah, "Localization of mobile nodes in wireless networks with correlated in time measurement noise," *Mobile Computing*, vol. 10, pp. 44–53, 2011.

[24] H. Miura, K. Hirana, N. Matsuda, H. Taki, N. Abe, and S. Hori, "Indoor localization for mobile node based on RSSI," in *Knowledge-Based Intelligent Information and Engineering Systems*, vol. 4694 of *Lecture Notes in Computer Science*, pp. 1065–1072, Springer, Berlin, Germany, 2007.

[25] M. Bertinato, G. Ortolan, F. Maran et al., "RF Localization and tracking of mobile nodes in wireless sensors networks: architectures, algorithms and experiment," Tech. Rep., University of Padova, Padua, Italy, 2008, http://paduaresearch.cab.unipd.it/1046/ .

[26] D. B. Faria, "Modeling signal attenuation in IEEE 820. 11 wireless LANS-Vol. 1, Kiwi Project," Tech. Rep. TR-KP06-0118, Stanford University, Stanford, Calif, USA, 2005.

[27] H. R. Jacobs, *Geometry*, Freeman, San Francisco, Calif, USA, 1987.

[28] Y. Ganggang, Y. Fengqi, and F. Lei, "A three dimensional localization algorithm using a mobile anchor node under wireless channel," in *International Joint Conference on Neural Networks (IJCNN '08)*, pp. 477–483, Neural Networks, June 2008.

[29] P. Chen, Z. Zhong, and T. He, "Bubble trace: mobile target tracking under insufficient anchor coverage," in *31st International Conference on Distributed Computing Systems (ICDCS '11)*, pp. 770–779, June 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 949285, 11 pages
http://dx.doi.org/10.1155/2013/949285

Research Article

A Location Predicting Method for Indoor Mobile Target Localization in Wireless Sensor Networks

Peng Gao,[1] Weiren Shi,[1] Wei Zhou,[2] Hongbing Li,[1] and Xiaogang Wang[1]

[1] *College of Automation, Chongqing University, Chongqing 400030, China*
[2] *School of Electrical and Information Engineering, Chongqing University of Science and Technology, Chongqing 401331, China*

Correspondence should be addressed to Peng Gao; jallenwill12@yahoo.cn

Received 6 September 2012; Revised 29 January 2013; Accepted 30 January 2013

Academic Editor: Wei Meng

Node position information is one of the important issues in many wireless sensor networks' usages. In this paper, based on path planning, a location predicting method (PPLP) for indoor mobile target localization is proposed. We first establish the path planning model to constrain the movement trajectory of the mobile target in indoor environment according to indoor architectural pattern. Then, one certain localization result can be obtained using MLE algorithm. After that, based on the path-planning model and some previous localization results, the most likely position of the target in the next time interval can be predicted with the proposed predicting approach. Finally, the MLE result and prediction result are weighted to obtain the final position. The simulation results demonstrate the effectiveness of the proposed algorithm.

1. Introduction

Wireless sensor networks (WSNs) are widely applied in monitoring, sensing, and collecting the information of interest in the environment [1]. Localization of target nodes is a fundamental problem in wireless sensor networks [2]. Up to now, the most existing localization algorithms of WSNs can be classified into two categories: range-based [3, 4] and range-free [5, 6]. Range-based algorithms use distance or angle estimates in their location estimations. Range-free algorithms use connectivity information between unknown nodes and anchor nodes. Range-based localization algorithms need to measure the actual distances or orientation between adjacent nodes, and then use the measured data to locate unknown nodes. Some ranging methods have been used for distance or orientation estimation, such as RSSI [7, 8], ToA [7, 9], TDOA [7, 10], and AoA [7, 11]. Whatever the ranging method is, there will be measurement errors in practical localization systems that result in noisy range estimations. Thus, accuracy in the position estimation phase is highly sensitive to range measurements [12]. Without improving range estimation or adding some other information related to localization, the accuracy of the current range-based algorithms cannot be improved obviously.

Indoor localization of WSNs has been a hot research topic for the last several years. Due to the randomness of targets moving and the complicated indoor environment, it is very different to locate indoor mobile target. In this paper, we proposed a location predicting method (PPLP) for indoor mobile target localization in WSNs based on path planning. We first establish the path-planning model to constrain the movement trajectory of the mobile target in indoor environment according to indoor architectural pattern. Then, we use MLE approach to get one certain location result of the target. After that, based on the path-planning model and some previous localization results of the target, the best possible position of the target in the next time interval can be predicted with the proposed predicting approach. Finally, the MLE result and prediction result are weighted to obtain the final position. In simulation process, we define three metrics to evaluate the performance of the proposed algorithm and compared with the MLE algorithm and PSO algorithm. Simulation results demonstrate that the proposed algorithm performed better than the other two algorithms.

The rest of this paper is organized as follows: in the next section, some related work is briefly introduced. Section 3 presents a detailed description of the main contribution of

this paper, the proposed algorithm PPLP. The simulation results on localization performance and error analysis are discussed in Section 4. Section 5 concludes.

2. Related Work

2.1. Maximum Likelihood Estimation. Maximum likelihood estimation (MLE) is widely used in many localization applications in wireless sensor networks [13–15]. In the localization process, the number of multiple measurement equations is usually more than the number of variables. Set r_i $(i = 1, 2, \ldots, n)$ is the estimated distance from anchor sensor node (x_i, y_i) to the target node, the target's position can be calculated as [16]:

$$\widehat{\mathbf{U}} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b}, \tag{1}$$

where

$$\mathbf{A} = 2\begin{bmatrix} x_n - x_1 & y_n - y_1 \\ x_n - x_2 & y_n - y_2 \\ \vdots & \vdots \\ x_n - x_{n-1} & y_n - y_{n-1} \end{bmatrix}, \qquad \widehat{\mathbf{U}} = \begin{bmatrix} x_t \\ y_t \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} \left(r_1^2 - r_n^2\right) - \left(x_1^2 - x_n^2\right) - \left(y_1^2 - y_n^2\right) \\ \left(r_2^2 - r_n^2\right) - \left(x_2^2 - x_n^2\right) - \left(y_2^2 - y_n^2\right) \\ \vdots \\ \left(r_{n-1}^2 - r_n^2\right) - \left(x_{n-1}^2 - x_n^2\right) - \left(y_{n-1}^2 - y_n^2\right) \end{bmatrix}. \tag{2}$$

2.2. Particle Swarm Optimization for Localization. Particle swarm optimization (PSO) [17, 18] is a swarm bionic optimization algorithm, which models the behavior of flocks of birds and fish. This method converges to the most optimal solution in a larger probability. Its process does not depend upon the quality of the objective function. So, it is commonly used to solve the optimization problems.

Let $x_i = (x_{i1}, x_{i2})$ be the 2-dimensional vector representing the position of the ith particle in the swarm, $g = [g_1, g_2]$ the position of the best particle in the swarm, $p = [p_{i1}, p_{i2}]$ the current best optimal solution of the ith particle itself and $v_i = [v_{i1}, v_{i2}]$ is the velocity of the ith particle. The particles evolve according to the following equations:

$$\begin{aligned} v_{id} &= \omega v_{id} + c_1 r_1 \left(p_{id} - x_{id}\right) + c_2 r_2 \left(g_d - x_{id}\right), \\ x_{id} &= x_{id} + v_{id}, \end{aligned} \tag{3}$$

where $d = 1, 2$; $i = 1, 2, \ldots, K$; K is the size of the swarm population; ω is the inertial weight; c_1 determines how much a particle is influenced by the memory of its best solution; and c_2 is an indication of the impact of rest of the swarm on the particle. c_1 and c_2 are termed cognitive and social scaling parameters, respectively. r_1 and r_2 are uniform random numbers in the interval $[0, 1]$.

Reference [18] proposed an improved PSO algorithm with RSSI self-correcting localization algorithm for wireless sensor networks. Based on the RSSI ranging, the author combined the proposed RSSI self-correction mechanism and improved PSO algorithm to optimize the nodes' localization for WSNs. Reference [12] proposed two novel and computationally efficient metaheuristic algorithms based on tabu search (TS) and particle swarm optimization (PSO) principles for locating the sensor nodes in a distributed wireless sensor network (WSN) environment. The author compared the performance of the proposed algorithms with each other and also against simulated annealing. The effects of range measurement error, anchor node density, and uncertainty in the anchor node position on localization performance are also studied through various simulations.

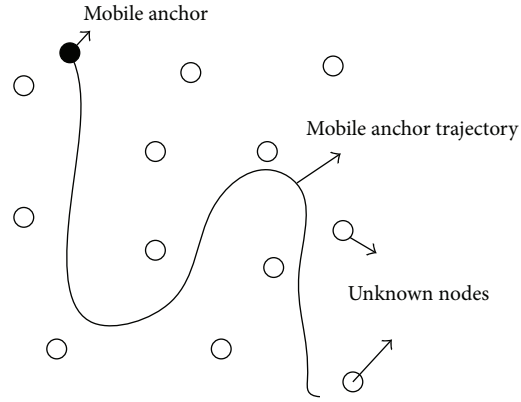


Figure 1: A mobile anchor assisting in the localization.

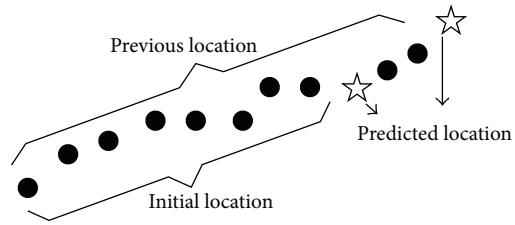


Figure 2: Prediction method for WSNs localization.

2.3. Path-Planning Method for WSNs Localization. Path planning is usually used for mobile anchor node in WSNs localization, where usually requires complex hardware support [19]. A mobile anchor node could be a small mobile robot equipped with a GPS and transmit its coordinate to the rest of the sensors to help them localize themselves. Figure 1 depicts a sensor network deployed over a geographical area. After the deployment, a mobile anchor traverses the sensor network while broadcasting its location packet. The packet contains the coordinates of the anchor, the current time, and some other information such as RSSI. Any node receiving the packet will be able to infer its location with several mobile anchors or one mobile anchor at different times.

2.4. Prediction Method for WSNs Localization. Prediction method is usually used to predict the possible locations of target in the next time interval based on the existing time series data [20] as Figure 2 shows.

Salamah and Doukhnitch [9] proposed a new efficient algorithm based on time of arrival (ToA) to determine the position of a mobile object (MO) in a wireless environment. However, it is not suitable for indoor mobile target localization because of the non-line-of-sight (NLOS) propagation in indoor environment.

3. Proposed Algorithm

3.1. Assumptions. We assume that the whole network consists of some stationary anchor nodes (ANs) and a mobile target. The anchor nodes whose coordinates are known are randomly or artificially deployed in a 2-dimensional indoor flat environment. All anchor nodes have the same radio transmission range (R). A mobile target may be a human, a robot, or some object manipulated by some person. Turning point (TP) is the intersection of two subpaths. The target can move freely among various rooms. After encountering some turning points, the target may change or not change its motion path. The position of the target can be calculated periodical with the proposed algorithm. The trajectory of the target can be regarded as a series of discrete points called target nodes (TNs). So, the localization problem changes into solving the locations of the target nodes.

3.2. Path-Planning Model. Generally, the movement of the mobile target (such as a person) is driving by its intention with large randomness. But in indoor environment, the motion trajectory of the target is relatively fixed because of the spatial constraint. People often engage in some typical motion patterns. For example, if a person wants to go to another nonadjacent room, he/she must go out the door first, then cross the corridors, and finally reach his/her destination. It is impossible for him/her to go through walls directly to reach the final position. Target's indoor movement will be limited by the indoor architecture pattern, such as walls and doors. Suppose the location system knows the indoor architectural pattern beforehand, and use it to assist positioning, we can get a better localization accuracy and trajectory of the target.

As Figure 3 shows, any corridor/aisle or room can be viewed as a path. Assume that each path can be described using function $f(x, y)$, then all possible moving paths can be described using path function as follows:

$$F(x,y) = \begin{cases} f_1(x,y), \\ f_2(x,y), \\ \quad\vdots \\ f_m(x,y), \end{cases} \quad x \in X; y \in Y, \tag{4}$$

where X and Y are the ranges of the x-coordinate and y-coordinate, respectively, and $f_m(x, y)$ is the path function for the mth path, called subpath function. All subpath functions form the total path function $F(x, y)$.

However, different buildings have different indoor architectural patterns. In order to make location computing more effective, we use a straight line segment function to describe each subpath as follows:

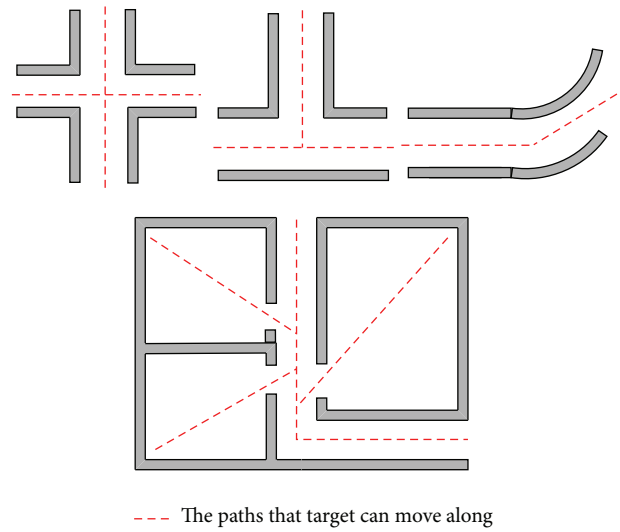


Figure 3: An indoor architectural pattern with some indoor paths.

$$(x_b - x_a)\, y = (y_b - y_a)\, x + (x_b y_a - y_b x_a), \tag{5}$$

$$\text{s.t.} \begin{cases} \min[x_a, x_b] \le x \le \max[x_a, x_b], \\ \min[y_a, y_b] \le y \le \max[y_a, y_b], \end{cases} \tag{6}$$

where (x_a, y_a) and (x_b, y_b) are the jumping-off point (JOP) and the end point (EP) of this staight line segment, respectively. A straight line segments subpath can be obtained once JOP and EP are determined. This function can completely (if the real subpath is straight) or approximately (if the real subpath is not straight) describe the real subpath. It will be useful to improve localization accuracy.

3.3. Location Predicting Method. We assume that the maximum velocity of human moving is $v_{\max}$, and localization is periodically with period being ΔT. It is difficult to determine TN's position according to the previous localization results, because the human moving is random and the localization error exists. However, the localization results can track target's trajectory with high possibility. So our strategy is, first, to compute localization results during a period of time T using some certain localization method (such as MLE); second, to predict the next possible positions according to these localization results; last, the localization result and prediction result are weighted to obtain the final position. In this paper, we use MLE algorithm to finish the first step. We only focused on step two and step three.

Let us use set $G = \{G_1, G_2, \ldots, G_k\}$ to describe localization results of the first step during time T, where $G_i = (x^{(i)}, y^{(i)})$. The prediction problem can be described as follows: how to get the next position $\widehat{G}_{k+1}$ according to set G and the path-planning model.

For any subpath $f(x, y)$, a set Z can be used to describe all points on this subpath. Each element of set Z satisfied

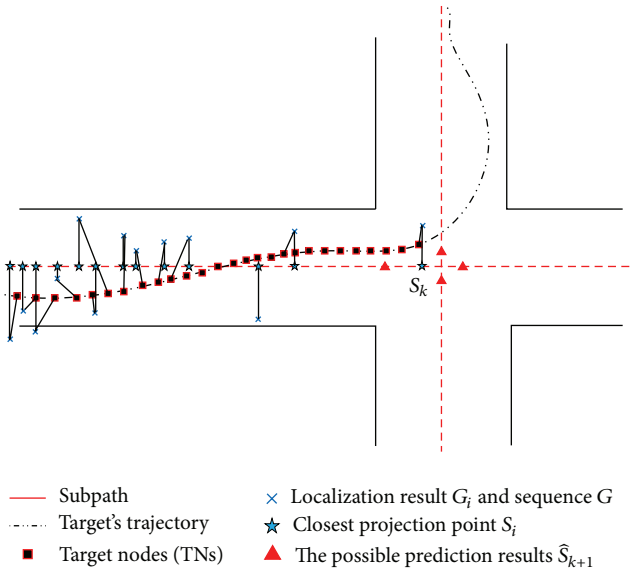


Figure 4: Predicting model in indoor environment.

function (5). We can also get that all elements of G are belong to set D which can be described as follows:

$$D = \left\{(x_D, y_D) \mid \begin{cases} x_{\min} - \Delta x \le x_D \le x_{\max} + \Delta x \\ y_{\min} - \Delta y \le y_D \le y_{\max} + \Delta y \end{cases}\right\}, \quad (7)$$

where $x_{\min}$, $y_{\min}$, $x_{\max}$, and $y_{\max}$ are the minimum x-coordinate, minimum y-coordinate, maximum x-coordinate, and maximum y-coordinate among all elements of G, respectively. Δx and Δy are threshold values which are related to accuracy of MLE algorithm.

One key point for predicting target's position is to find which subpath the target may move at time k. Some definitions are defined at first as follows:

Definition 1. Optional subpath that target may move on: for any subpath $f(x, y)$ described with set Z, if it is satisfied that $Z \cap D \neq \emptyset$, then this subpath is one optional subpath.

Definition 2. Closest projection point S_i and set S: S_i is the closest projection point of G_i which satisfies the following function (8), S is the set of $\{S_1, S_2, \ldots, S_k\}$ whose element number is equal to G's as

$$S_i = \left\{\widehat{S}_x \mid \widehat{S}_x = \arg\min \left|S_x - G_i\right|\right\}, \quad (8)$$

where $S_x = (x_s, y_s)$ is any point on subpath $f_i(x, y)$. $f_i(x, y)$ is one of all optional subpaths.

The prediction model can be showed as in Figure 4.

Usually, the subpath $f_k(x, y)$ in which S_k is on is the most possible subpath that target may move on. In order to increase the predicting probability, we choose the subpath most of S_{k-p} to S_k are on as the kth subpath that targets moves on. Here p is constant which is determined by experiment. p should be satisfied during time $k - p$ to k; the distance of target's moving is small. Then, we use the closet projection points on $f_k(x, y)$ to form a new set $S' = \{S^{(1)}, S^{(2)}, \ldots, S^{(k)}\}$. And the prediction problem based on the previous model can be written as follows:

$$\widehat{S}_{k+1} = S^{(k)} + v_k \cdot \Delta T, \quad (9)$$

where $\widehat{S}_{k+1}$ is the position to be predicted, and v_k is the velocity at time k. For the randomness of target moving, the direction of vector v_k is hard to be determined. So, we rewrite it as follows:

$$\left[\!\left[\widehat{S}_{k+1}, S^{(k)}\right]\!\right] = \left|v_k\right| \cdot \Delta T = \Delta S, \quad (10)$$

where $[\![\cdot, \cdot]\!]$ denotes the shortest distance from one point to anther along some certain subpath. So, $[\![\widehat{S}_{k+1}, S^{(k)}]\!]$ is the shortest distance from $S^{(k)}$ to $\widehat{S}_{k+1}$ along some certain subpath.

Obviously, the optional subpath targets are on at time $k + 1$ is very likely more than one. So, $\widehat{S}_{k+1}$ may have one or multiple solutions. At time $k+1$, target may still be on subpath $f_k(x, y)$ or turn to another adjacent subpath. Without loss of generality, we assume that $f_{k+1}(x, y)$ is the possible subpath at time $k+1$. The key point to judge whether $f_{k+1}(x, y)$ exist is to find out whether there is a TP when target is moving ahead during time ΔT.

Let C be the set of all possible TPs that target may encounter, if there is a point C_r in C satisfing (10), then $f_{k+1}(x, y)$ exists as follows:

$$\left[\!\left[C_r, S^{(k)}\right]\!\right] < \Delta S, \quad (11)$$

where $[\![C_r, S^{(k)}]\!]$ is the possible shortest distance from $S^{(k)}$ to C_r along the subpath $f_k(x, y)$ which can be obtained by as following:

$$\left[\!\left[C_r, S^{(k)}\right]\!\right] = \int_{S^{(k)}}^{C_r} f_k(x, y). \quad (12)$$

Then the set of all possible predicting positions at time $k + 1$ can be written as follows:

$$M_{k+1} = \left\{Q_r \mid \Delta S = \begin{cases} \int_{S^{(k)}}^{C_r} f_k(x, y) + \int_{C_r}^{Q_r} f_{k+1}(x, y), \\ \qquad \text{if } \left[\!\left[C_r, S^{(k)}\right]\!\right] < \Delta S \\ \int_{S^{(k)}}^{Q_r} f_k(x, y), \text{ else} \end{cases}\right\}. \quad (13)$$

Set M_{k+1} contains all possible predicting positions. But the possibility of each element in M_{k+1} to become the final localization result is different. Let $\widehat{\mathbf{U}}_{k+1}$ be the localization result using MLE algorithm at time $k + 1$. Generally, $\widehat{\mathbf{U}}_{k+1}$ is close to real position with high possibility. The more accurate the MLE is, the higher the possibility will be. And the element in M_{k+1} nearby $\widehat{\mathbf{U}}_{k+1}$ has a higher possibility than the other elements. Predicting result in M_{k+1} that owns this feature can be treated as one final result, that is:

$$\widehat{S}_{k+1}^{(a)} = \left\{M_j \mid \min_{M_j \in M_{k+1}} \left|M_j - \widehat{\mathbf{U}}_{k+1}\right|\right\}. \quad (14)$$

On the other hand, for the randomness of human moving, different movement patterns may lead to different prediction possibilities. We can infer the next possible positions according to previous locations.

Definition 3. Direction value of $S^{(i)}$: for the ith point $S^{(i)}$ in S', we use $\delta_{\text{orien}}(S^{(i)} \mid S^{(i-1)})$ to describe the target's moving direction at time i. If $\delta_{\text{orien}}(S^{(i)} \mid S^{(i-1)})$ equals to 1, the moving direction of $S^{(i)}$ is forward, otherwise backward. $\delta_{\text{orien}}(S^{(i)} \mid S^{(i-1)})$ can be calculated with the following:

$$\delta_{\text{orien}}\left(S^{(i)} \mid S^{(i-1)}\right) = \begin{cases} 1, & \text{if } < \vec{\psi}, \vec{l} \geq 0, \text{where}, \vec{\psi} = S^{(i)} - S^{(i-1)}, \\ & \vec{l}\left(\left.\frac{\partial f}{\partial x}\right|_{S^{(i)}}, \left.\frac{\partial f}{\partial y}\right|_{S^{(i)}}\right), \\ -1, & \text{else}, \end{cases} \tag{15}$$

where $f = f_k(x, y), i \geq 2$.

So the possibility the target moving forward at time k can be described as follows:

$$p_{(k+1|k)}^{(\text{forward})} = \frac{\text{Num\_}\delta^{(+1)}}{m}, \tag{16}$$

where Num_$\delta^{(+1)}$is the amount of points whose direction value is 1; m is the amount of elements in S'.

From the previous description, we know that a TP may be encountered when target is moving forward or backward if $f_{k+1}(x, y)$ exists. So, in set M_{k+1}, some elements may reflect the prediction results that target moving forward; we use $n^{(\text{forward})}$ to denote the number of these elements. Others may reflect the prediction results that target moving backward; we use $n^{(\text{backward})}$ to denote the number of these elements. So, we can get another prediction result:

$$\widehat{S}_{k+1}^{(b)} = \sum_{i=1}^{n^{(\text{forward})}} \left(\frac{p_{(k+1|k)}^{(\text{forward})}}{n^{(\text{forward})}} \cdot \widehat{Q}_{k+1}^{(i)}\right) + \sum_{j=1}^{n^{(\text{backward})}} \left(\frac{1 - p_{(k+1|k)}^{(\text{forward})}}{n^{(\text{backward})}} \cdot \widehat{Q}^{(j)}{}_{k+1}\right), \tag{17}$$

where $\widehat{Q}_{k+1}^{(i)}$ is the ith prediction result in set M_{k+1} when target is moving forward, $\widehat{Q}_{k+1}^{(j)}$ is the jth prediction result in set M_{k+1} when backward.

Then the final predication result can be written as follows:

$$\widehat{S}_{k+1} = \alpha \cdot \widehat{S}_{k+1}^{(a)} + (1 - \alpha) \cdot \widehat{S}_{k+1}^{(b)}, \tag{18}$$

where α is the weight of each predicting result. It can be obtained with some learning methods [20] when doing long-term prediction [21] applications. Generally, the long-term motion trajectory of the target usually comply with limited movement patterns, which is shown as a repetitive motion along one or several paths. In this paper, we only consider short-term predicting, and the value of α is set to 0.5.

3.4. Final Location Computing. The final localization result can be obtained as follows:

$$G_{k+1} = w \cdot \widehat{\mathbf{U}}_{k+1} + w' \cdot \widehat{\mathbf{S}}_{k+1}. \tag{19}$$

Here we defined w as follows:

$$w = \begin{cases} \dfrac{\left|\left|\nu_{\max}\right| \cdot \Delta T - \left[\!\left[S^{(k)}, \widehat{S}_{k+1}\right]\!\right]\right|}{\left|\widehat{S}_{k+1} - \widehat{\mathbf{U}}_{k+1}\right|}, \\ \text{if } \left|\widehat{\mathbf{U}}_{k+1} - S^{(k)}\right| \geq \left|\nu_{\max}\right| \cdot \Delta T, \\ 1, \qquad \text{else.} \end{cases} \tag{20}$$

3.5. Updating Rules. After getting the localization result at time $k + 1$, some updating rules are proposed for the next location predicting and computing. The updating rule for $|\nu_{k+1}|$ can be written as follows:

$$\left|\nu_{k+1}\right| = \frac{\left[\!\left[S_{k+1}, S_k\right]\!\right]^{(f_k(x,y))}}{\Delta T}, \quad \text{if } \left(\left|\nu_{k+1}\right| < \left|\nu_{\max}\right|\right), \qquad \left|\nu_{k+1}\right| = \left|\nu_{\max}\right|, \quad \text{else}, \tag{21}$$

where $\left[\!\left[S_{k+1}, S_k\right]\!\right]^{(f_k(x,y))}$ is the distance from S_k to S_{k+1} along the subpath $f_k(x, y)$; S_{k+1} is the closest projection point of G_{k+1} which can be obtained by Definition 2. $f_{k+1}(x, y)$ is the subpath that S_{k+1} is on at time $k + 1$.

The update rule for set S is to keep the length of S unchanged, remove the first element, and insert the new element S_{k+1} into the last of S.

3.6. Pseudoprocedure of PPLP Algorithm. Target's final location at time $k + 1$ can be obtained with the following pseudo procedure of PPLP algorithm.

Procedure begin:

//Path-planning begin:

(1) input JOP and EP of each subpath according to actual indoor environment;

(2) determine each subpath with function (5);

(3) determine the total path with function (4);

complete path-planning modeling.

//Path-planning end.

(4) calculate the target's location $\widehat{\mathbf{U}}_{k+1}$ at time $k + 1$ using MLE algorithm;

//Predicting begin:

(5) initialize G, $|\nu_k|$, $|\nu_{\max}|$, p, Δx, Δy;

(6) calculate set S with the path-planning model and functions (7) and (8);

(7) calculate M_{k+1} with functions (9)–(13);

(8) calculated $\widehat{S}_{k+1}^{(a)}$with function (14);

(9) calculated $\widehat{S}_{k+1}^{(b)}$with function (15)–(17);

(10) calculated $\widehat{S}_{k+1}$ with function (18);

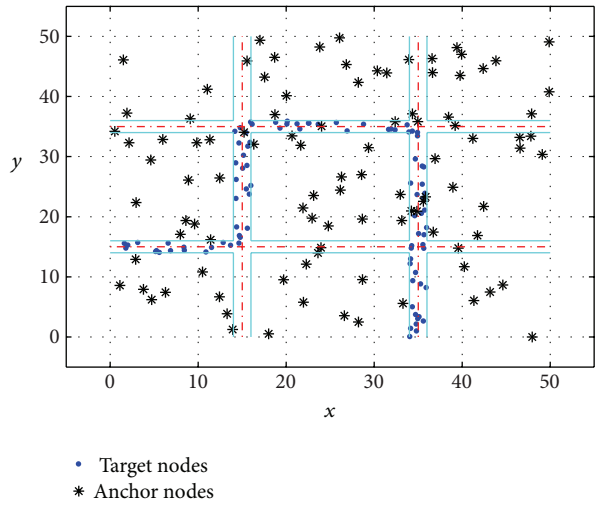


Figure 5: The nodes deployment and environment set up.

(11) calculated the final position G_{k+1} with function (19) and (20).

//Predicting end.

//Updating begin:

(12) update some with updating rules in Section 3.5.

//Updating end.

Procedure end.

4. Simulation and Analysis

In this section, we will evaluate the performance of the proposed localization algorithm through extensive simulations carried out using MATLAB.

4.1. Simulation Scenario and Settings. We set simulation scenario and some key parameters as follows.

All ANs are randomly deployed in a $50 * 50$ m^2 area for the simulation. The total number of ANs is initially 100 and every AN is known its position. All members of G are initialized to (0, 15). Some other parameters are shown in Table 1.The nodes deployment and the environment setup are shown in Figure 5.

In Figure 5, we use 4 dotted-line segments to represent 4 subpaths, respectively. The path width is set to be 2 m. We use some random discrete TNs (as shown in Figure 5 with blue dots) to simulate the randomness of human movement. In the proposed algorithm, we did not consider any particular ranging technique. In the simulation process, we use the following formula (22) [12, 22] to describe the measured distances between TNs and ANs with some certain ranging technique:

$$\widehat{d}_{ij} = d_{ij} + N_{ij}, \tag{22}$$

where $\widehat{d}_{ij}$ and d_{ij} are the measured and real distance between the AN$_i$ and the TN$_j$, respectively, and N_{ij} is assumed to be blurred by additive Gaussian random variables with zero mean and known variance σ_d^2.

Table 1: Simulation parameters.

Type	Value
p	5
$\lvert \mathbf{v}_k \rvert$	0.5
$\lvert \mathbf{v}_{\max} \rvert$	5 m/s
T	2 s
ΔT	100 ms
Δx	3 m
Δy	3 m
R	10 m

4.2. Evaluation Metrics. To analyze the simulation results, in this paper, we defined the following two metrics to evaluate the performance of the proposed algorithm.

(a) Average localization error:

$$\text{err\_aver} = \frac{1}{\text{NUM}} \sum_{i=1}^{\text{NUM}} \left\| X_i - \sigma_i \right\|, \tag{23}$$

where err_aver is the error mean of localization result which reflects the accuracy of the algorithm. X_i is the real coordinate of the TN$_i$; σ is the calculated coordinate of the TN$_i$ using the proposed localization algorithm. $\|X_i - \sigma_i\|$ represents the localization error of TN$_i$. NUM is the number of TNs.

(b) Standard variance of localization error:

$$\text{Loc\_ver} = \sqrt{\frac{1}{\text{NUM}} \sum_{i=1}^{\text{NUM}} \left(\left\| X_i - \sigma_i \right\| - \text{err\_aver} \right)^2}, \tag{24}$$

where Loc_var is the standard variance of localization results which can describe the degree of spread of the localization results. Other variables have the same meanings as metric (a).

(c) Average distance to the correct subpath:

$$\text{deviate\_value\_aver} = \frac{1}{\text{NUM}} \sum_{j=1}^{\text{NUM}} \left\| \chi_j - \sigma_j \right\|, \tag{25}$$

where deviate_value_aver is the average distance that the location results of the targets deviated from the correct subpath. χ_j is the closest projection point of the TN$_j$, σ_j is the calculate coordinate of TN$_j$ using proposed algorithm localization algorithm. $\|\chi_j - \sigma_j\|$ represents the distance that TN$_i$ departed from the correct subpath.

4.3. Simulation Results and Analysis. We firstly simulate 100 ANs to evaluate the performance of the proposed algorithm and the classical MLE algorithm [13–16]. The simulation results are shown in Figures 6 and 7.

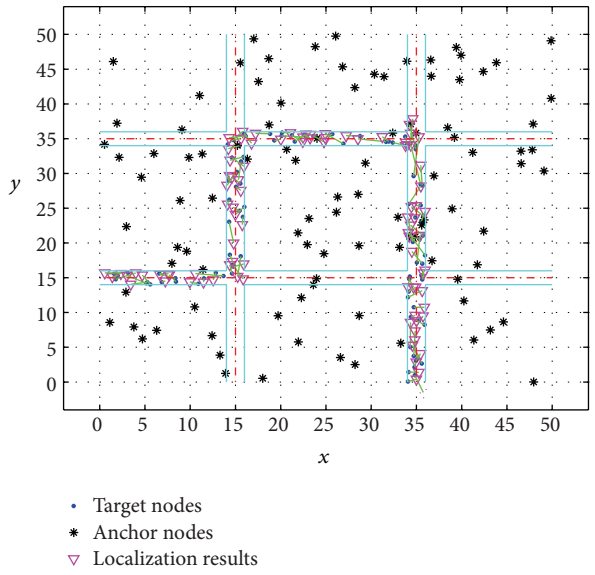


Figure 6: The simulation result of the proposed algorithm.

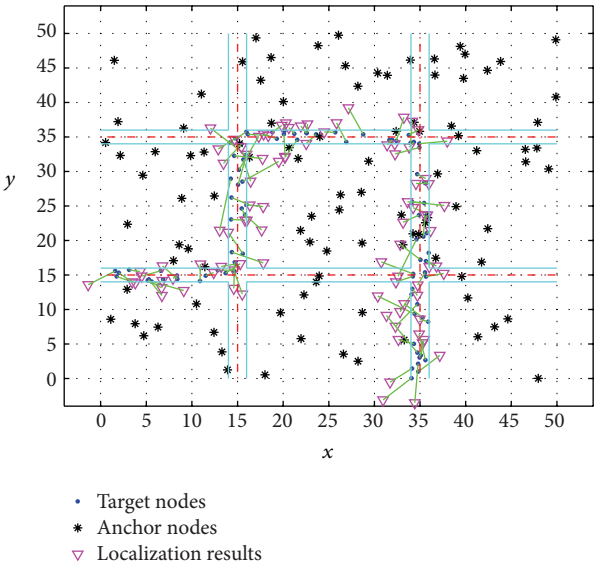


Figure 7: The simulation result of MLE algorithm.

Figures 6 and 7 show the simulation results of the proposed algorithm and MLE algorithm when the number of anchor nodes is 100 and transmission range (R) is 10 m. We can see that the performance of the proposed algorithm is better than MLE algorithm. To ease the understanding and analyzing of simulation results, we use average localization error (err_aver), standard variance of localization error (Loc_var), and average distance to the correct subpath (deviate_value_aver) as the evaluation metrics to evaluate the performance of these two algorithms. Finally, we get the following comparison results.

Figure 8 provides an intuitive comparison of the accuracy of the proposed localization algorithm and the MLE. The average localization error can be obtained using formula (23). The result shows that the average localization error of MLE is 2.5621 m while the proposed algorithm is only 1.7624 m. We can see that the proposed localization algorithm has a better accuracy than MLE algorithm. Figure 9 shows the distance that localization results of TNs deviated from the correct trajectory when the target moves along the correct subpath as shown in Figure 5. The average distance to the correct subpath can be obtained using formula (25). The simulation result shows that the average distance to the correct subpath of the proposed algorithm is 0.5175 m, which is much smaller than MLE algorithm. We also calculate the standard variance of localization error with formula (24). The Loc_var of MLE is 1.4318, and the proposed algorithm's is 1.0972. Figure 10 gives the comparison results of these two algorithms with respect to the proposed three metrics. The result shows that the proposed algorithm (PPLP) has better performance in all evaluation metrics than MLE algorithm. The accuracy is high, the localization result is stable and concentrated, and it can always find the right way that the target moves on. This is very useful in some practical applications such as elders/children guarding, hospital patients care, indoor searching, and rescuing for trapped.

In order to further verify the effectiveness of the proposed algorithm, we also did some extensive simulations and compared it with the PSO algorithm [12, 17, 18]. By changing the transmission radius (R) and anchor nodes ratio, we get the following simulation results.

Figure 11 provides a comparison of the accuracy of the proposed localization approach, the MLE algorithm, and the PSO algorithm with respect to anchor nodes' ratio and average connectivity. We run the simulation with 90 TNs. The number of anchor nodes varied from 30 to 100 (as a result the average connectivity increased from 3.99 to 11.73). The simulation results show that the proposed algorithm has a higher accuracy than the other two algorithms. Figure 12 shows the results of these three algorithms with respect to the standard variance of localization error when the number of anchor nodes was changed from 30 to 100. The results show that the standard variance of localization error of the proposed algorithm is lower than the other two algorithms. Figure 13 gives the simulation results of average distance to the correct subpath when the simulation setting is the same as Figure 11. After running at least 100 times simulations, the average distance to the correct subpath can be obtained. As can be seen from Figure 13, it is obvious that the average distance to the correct subpath decreases when anchor nodes' ratio increase. But simulation result of the proposed algorithm changed within a narrow range from 0.51 m to 0.86 m, while the other two algorithms changed obviously. That is to say, the proposed algorithm is more stable than the other two algorithms in indoor environment.

Some more simulation results about the discussed three metrics of these three algorithms can be observed in Figures 14, 15, and 16. We run the simulation with 90 TNs and 100 ANs. The transmission range was increased from 5 m to 15 m (as a result, the average connectivity increased from 3.15 to 16.27). The transmission range of a sensor node varies with its transmission power. A better localization performance is expected with higher transmission range as the number

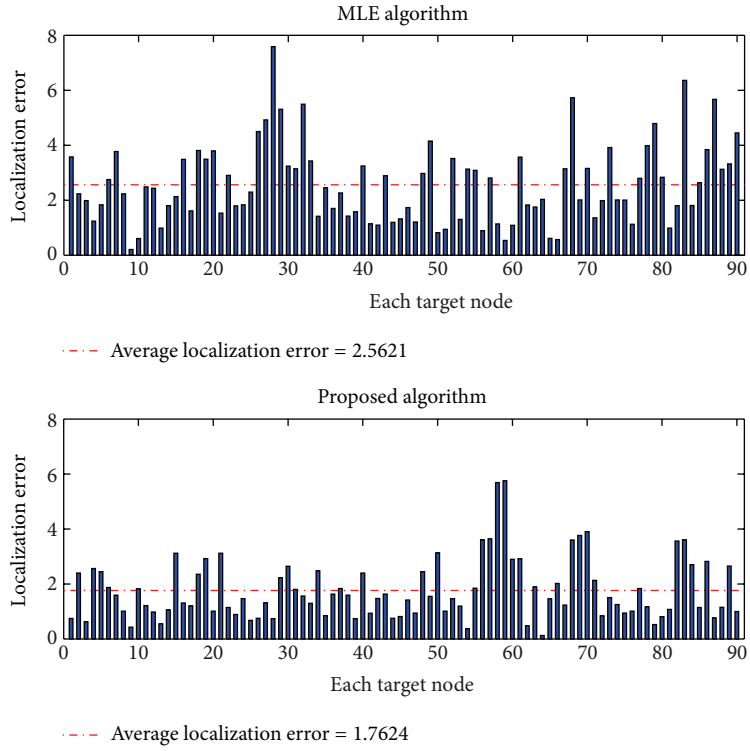


Figure 8: Accuracy comparison between the proposed algorithm and MLE.

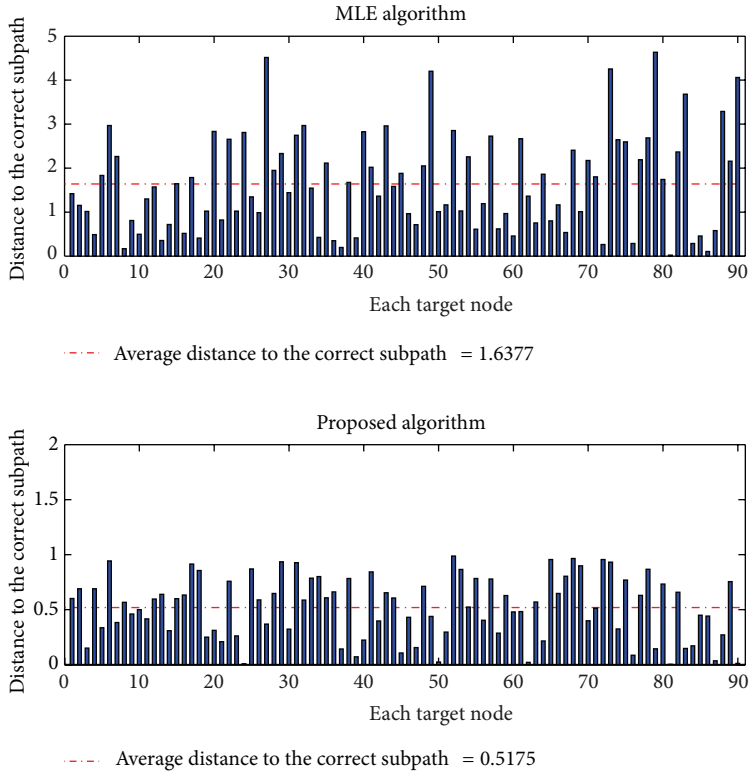


Figure 9: Distance to the correct subpath comparison between the proposed algorithm and MLE.

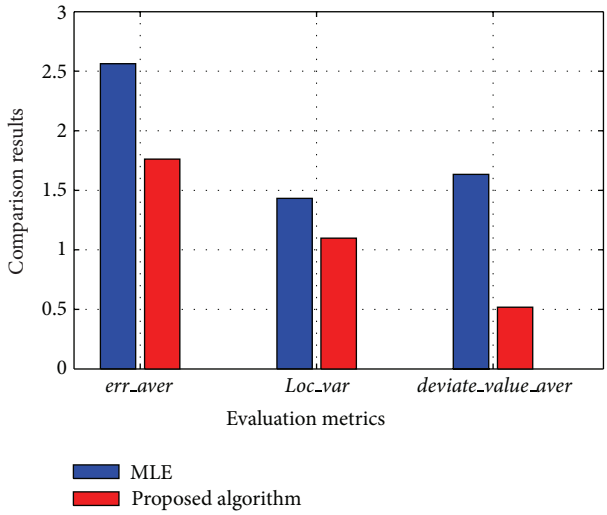


Figure 10: Comparison results of MLE and the proposed algorithm with respect to the proposed three evaluation metrics.

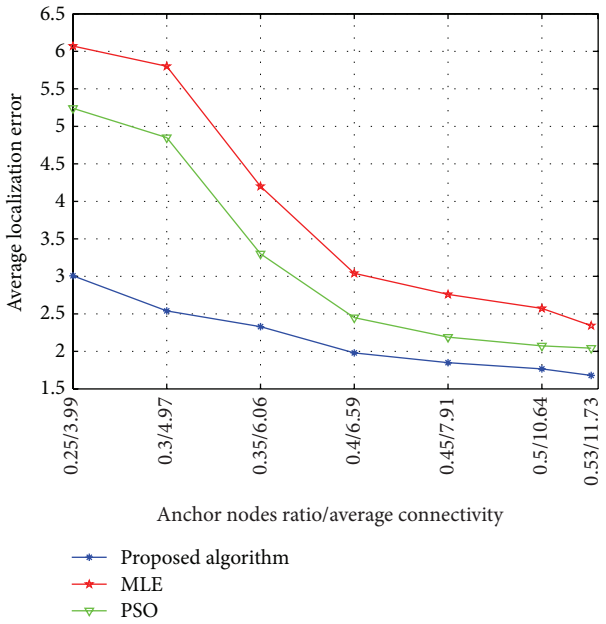


Figure 11: Average localization error versus anchor nodes' ratio/Average connectivity.

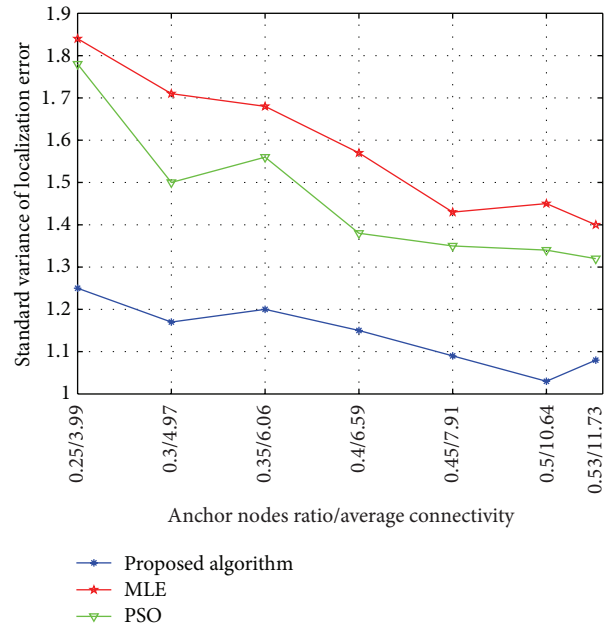


Figure 12: Standard variance of localization error versus anchor nodes' ratio/average connectivity.

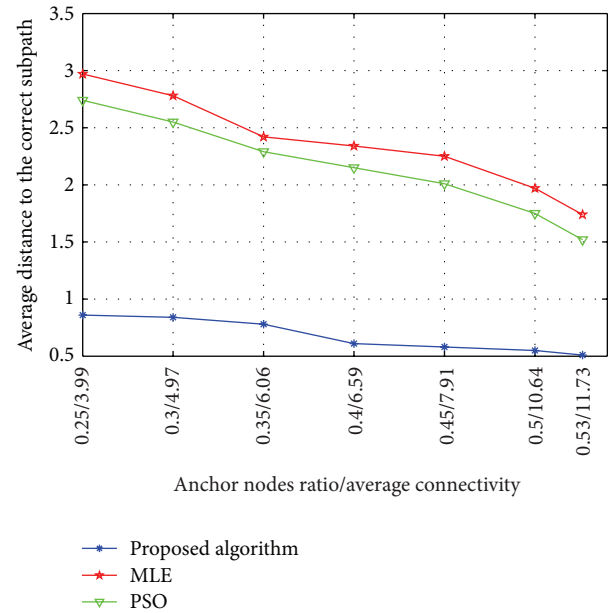


Figure 13: Average distance to the correct subpath versus anchor nodes' ratio/average connectivity.

of one-hop ANs increases [14]. With the increase in transmission range, the average localization error, the standard variance of localization error, and the average distance to the correct subpath decrease. But the decrease scopes of these three metrics are not all obvious when transmission range is larger than 10 m (here the connectivity value is 11.60). That is because the connectivity is an essential factor that affects these algorithms' performance. When connectivity is greater than a certain value (such as 11.60 showed in Figures 14–16), the accuracy of the algorithm is changed little. Before that, connectivity can greatly affect algorithm's performance. However, the proposed algorithm can have an excellent performance even with low connectivity as showed in Figures 14–16.

All simulation results proved that PPLP algorithm is performed well in indoor environment. Furthermore, PPLP algorithm is a centralized computing method. The location calculation of the target can be done in some certain device with strong processing capacity such as personal computer. The proposed algorithm does not need to calculate large matrix; there is no iteration in localization computing,

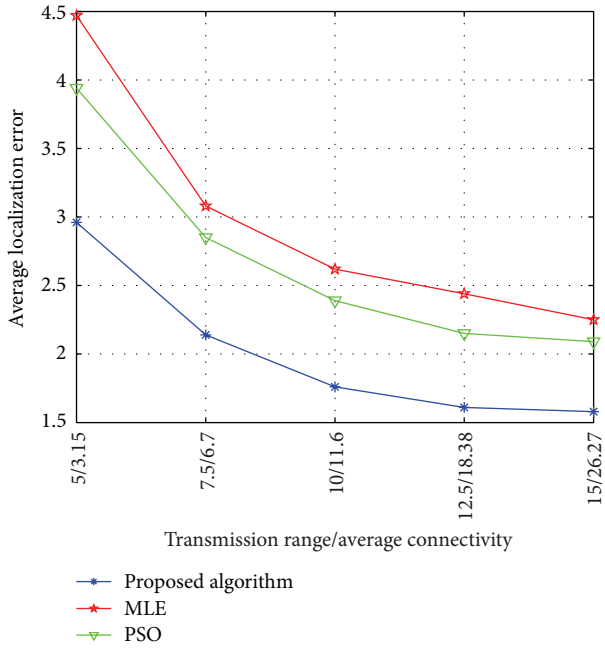


Figure 14: Average localization error versus transmission range/average connectivity.

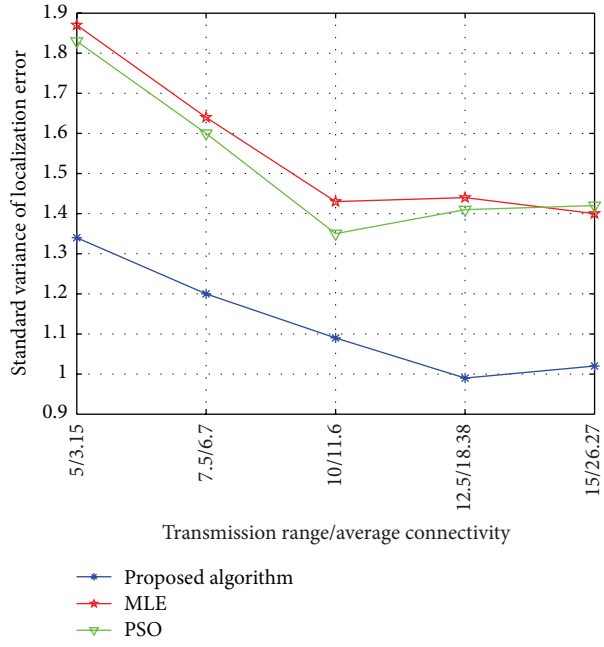


Figure 15: Standard variance of localization error versus transmission range/average connectivity.

the localization program is executed sequentially with high efficiency and low complexity.

5. Conclusion

Localization is one of the substantial issues in wireless sensor networks. In this paper, we presented a location predicting method (PPLP) for indoor mobile target localization in WSNs based on path-planning. We first analyzed the common feature of indoor environment for most buildings and the motion pattern of most targets and established the path-planning model to constrain the movement trajectory of the mobile target according to indoor architectural pattern. Then, we used MLE algorithm to obtain one certain kind of location result of the target. After that, based on the path-planning model and some previous localization results of the target, the best possible position of the target in the next time interval was predicted with the proposed predicting approach. Finally, the MLE result and prediction result were weighted to obtain the final position. In simulation process, we defined three metrics to evaluate the performance of the proposed algorithm and compared it with the MLE algorithm and PSO algorithm. Simulation results showed that the proposed algorithm has a better performance in all these three evaluation indicators and can be very useful for some practical applications such as elders/children guarding, hospital patients care, indoor search, and rescue for trapped.

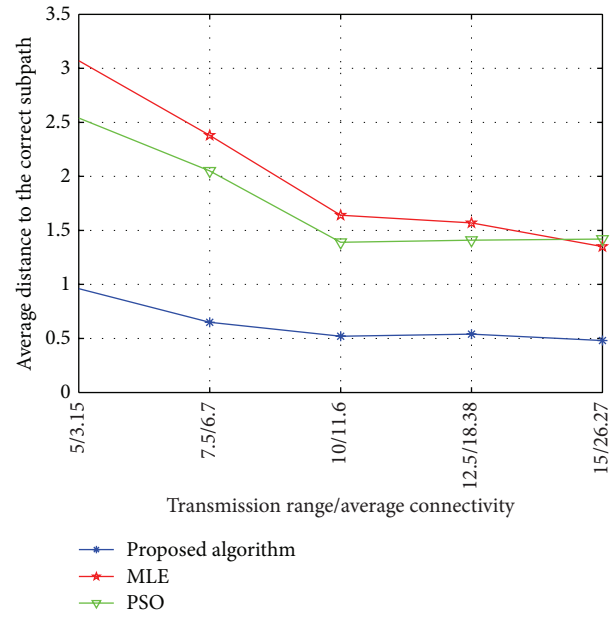


Figure 16: Average distance to the correct subpath versus transmission range/average connectivity.

Acknowledgments

This work was supported by the National Key Technology R&D Program (2011BAK07B03), National Science and Technology Major Project (2009ZX07528-003-09), and the 2011 Internet of Things Development Special Fund.

References

[1] T. Zhang, J. S. He, and H. Yu, "Secure localization in wireless sensor networks with mobile beacons," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 732381, 11 pages, 2012.

[2] J. Wang, R. K. Ghosh, and S. K. Das, "A survey on sensor localization," *Journal of Control Theory and Applications*, vol. 8, no. 1, pp. 2–11, 2010.

[3] H. Chen, P. Huang, M. Martins, H. C. So, and K. Sezaki, "Novel centroid localization algorithm for three-dimensional wireless sensor networks," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, pp. 1–4, Dalian, China, October 2008.

[4] K. Z. Liu, S. Wang, F. Zhang, F. P. Hu, and C. C. Xu, "Efficient localized localization algorithm for wireless sensor networks," in *Proceedings of the 5th International Conference on Computer and Information Technology (CIT '05)*, pp. 517–523, Shanghai, China, September 2005.

[5] S. Jian, L. Linlan, C. Yubin, and H. Haitao, "A novel three-dimensional localization algorithm in wireless sensor networks," in *Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, pp. 24–29, Dalian, China, October 2008.

[6] S. Zheng, L. Kai, and Z. H. Zheng, "Three dimensional localization algorithm based on nectar source localization model in wireless sensor network," *Application Research of Computers*, vol. 25, no. 8, pp. 2512–2513, 2008.

[7] K. Z. Lu, X. H. Xiang, D. Zhang, R. Mao, and Y. H. Feng, "Localization algorithm based on maximum a posteriori in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 260302, 7 pages, 2012.

[8] M. S. Rahman, Y. Park, and K. D. Kim, "RSS-based indoor localization algorithm for wireless sensor network using generalized regression neural network," *Arabian Journal For Science and Engineering*, vol. 34, no. 4, pp. 1043–4053, 2012.

[9] M. Salamah and E. Doukhnitch, "An efficient algorithm for mobile objects localization," *International Journal of Communication Systems*, vol. 21, no. 3, pp. 301–310, 2008.

[10] L. Girod and D. Estrin, "Robust range estimation using acoustic and multimodal sensing," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 1312–1320, Maui, Hawaii, USA, October 2001.

[11] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AoA," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies*, vol. 3, pp. 1734–1743, San Francisco, Calif, USA, March 2003.

[12] A. Gopakumar and L. Jacob, "Performance of some metaheuristic algorithms for localization in wireless sensor networks," *International Journal of Network Management*, vol. 19, no. 5, pp. 355–373, 2009.

[13] M. M. Noel, P. P. Joshi, and T. C. Jannett, "Improved maximum likelihood estimation of target position in wireless sensor networks using particle swarm optimization," in *Proceedings of the 3rd International Conference on Information Technology: New Generations (ITNG '06)*, pp. 274–278, Las Vegas, Nev, USA, April 2006.

[14] A. J. Weiss and J. S. Picard, "Maximum likelihood localization of wireless networks using biased range measurements," in *Proceedings of the International Symposium on Communications and Information Technologies*, pp. 865–870, Sydney, Australia, October 2007.

[15] Y. T. Chan, H. Y. C. Hang, and P. C. Ching, "Exact and approximate maximum likelihood localization algorithms," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 1, pp. 10–16, 2006.

[16] X. Chen and B. L. Zhang, "Improved DV-Hop node localization algorithm in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 213980, 7 pages, 2012.

[17] J. J. Yao, J. Li, L. Wang, and Y. Ha, "Wireless sensor network localization based on improved particle swarm optimization," in *Proceedings of the International Conference on Computing, Measurement, Control and Sensor Network (CMCSN '12)*, pp. 72–75, Taiyuan, China, July 2012.

[18] M. Y. Shen, Y. J. Lu, and M. S. Zhao, "Study of node localization algorithm based on improved particle swarm optimization and RSSI for WSNs," in *Informatics in Control, Automation and Robotics*, vol. 2 of *Lecture Notes in Electrical Engineering*, pp. 185–193, 2011.

[19] H. Li, J. Wang, X. Li, and H. Ma, "Real-time path planning of mobile anchor node in localization for wireless sensor networks," in *Proceedings of the IEEE International Conference on Information and Automation (ICIA '08)*, pp. 384–389, Zhangjiajie, China, June 2008.

[20] M. Jafari, N. Abdollahi, and H. Mohammadi, "Predicating the location of nodes in ad hoc network by lazy learning method," *Communications in Computer and Information Science*, vol. 241, pp. 336–345, 2011.

[21] M. J. Jahromi, A. I. Maswood, and K. J. Tseng, "Long term prediction of tidal currents," *IEEE Systems Journal*, vol. 5, no. 2, pp. 146–155, 2011.

[22] E. Niewiadomska-Szynkiewicz and M. Marks, "Optimization schemes for wireless sensor network localization," *International Journal of Applied Mathematics and Computer Science*, vol. 19, no. 2, pp. 291–302, 2009.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 529489, 10 pages
http://dx.doi.org/10.1155/2013/529489

Research Article

ALRD: AoA Localization with RSSI Differences of Directional Antennas for Wireless Sensor Networks

Jehn-Ruey Jiang, Chih-Ming Lin, Feng-Yi Lin, and Shing-Tsaan Huang

Department of Computer Science and Information Engineering, National Central University, Jhongli 32001, Taiwan

Correspondence should be addressed to Jehn-Ruey Jiang; jrjiang@csie.ncu.edu.tw

Received 30 September 2012; Revised 22 January 2013; Accepted 1 February 2013

Academic Editor: Chengdong Wu

In this paper, we fit RSSI values into a parabola function of the AoA between 0° and 90° by applying quadratic regression analysis. We also set up two-directional antennas with perpendicular orientations at the same position and fit the difference of the signal RSSI values of the two antennas into a linear function of the AoA between 0° and 90° by linear regression analysis. Based on the RSSI-fitting functions, we propose a novel localization scheme, called AoA Localization with RSSI Differences (ALRD), for a sensor node to quickly estimate its location with the help of two beacon nodes, each of which consists of two perpendicularly orientated directional antennas. We apply ALRD to a WSN in a 10 × 10 m indoor area with two beacon nodes installed at two corners of the area. Our experiments demonstrate that the average localization error is 124 cm. We further propose two methods, named maximum-point minimum-diameter and maximum-point minimum-rectangle, to reduce localization errors by gathering more beacon signals within 1 s for finding the set of estimated locations of maximum density. Our results demonstrate that the two methods can reduce the average localization error by a factor of about 29%, to 89 cm.

1. Introduction

A wireless sensor network (WSN) consists of tiny sensor nodes equipped with computational, communication, and sensing capabilities, whereby each sensor node can collect data about the environment, such as temperature, vibration levels, light, electromagnetic strength, and humidity. The sensed data is then transmitted to the sink node through a chain of multiple intermediate nodes that help forward the data. Due to their capabilities and versatility, WSNs have been widely used in many areas, such as military affairs, healthcare, and environmental monitoring. In many applications, apart from sensed data, the location information of the deployed sensor node is also desirable as it can be used to improve routing efficiency. Hence, the discovery of the locations or positions of sensor nodes is one of the most critical issues for WSNs.

Localization is the process of determining the absolute or relative physical location of a specific node or the *target node*. Although a global positioning system (GPS) [1] can provide precise location information, the costly hardware and large size make it unsuitable for WSNs. Furthermore, a GPS can only be used outdoors since it depends on signals directly received from satellites for localization. Besides the GPS, numerous localization methods [2–27] have also been proposed. Most of these deploy some *beacon* (or *anchor*) *nodes*, which periodically broadcast beacon signals containing their own locations to help other sensor nodes with the localization.

Localization schemes can be classified as range based or range-free. In range-free schemes, the sensor node location is estimated solely on network connectivity. Such schemes need no extra hardware, but their accuracy is too low, and they usually rely on a large deployment of beacon nodes to improve the accuracy. Conversely, range-based schemes usually have better accuracy. They measure the time of arrival (ToA) [2, 3], time difference of arrival (TDoA) [4–6], angle of arrival (AoA) [2, 4, 15, 17, 18, 20, 21, 27], and received signal strength indicator (RSSI) [3, 7–14, 16, 20, 21] to estimate the distances or angles between pairs of nodes, which in turn are used to calculate the locations of nodes. Most kinds of measurement are taken with extra auxiliary hardware. For example, ToA and TDoA are very sensitive to timing errors, and, hence, their measurement relies on highly accurate synchronized timers. The AoA, which is defined as the angle

between the propagation direction of an incident RF wave and a reference direction, can be measured by an array of antennas. Unlike the previously mentioned three kinds of measurement, RSSI can be outputted by most commercial off-the-shelf sensor nodes.

RSSI-based localization methods can be further classified into groups, such as propagation model [8, 9], proximity [10–12] or fingerprinting [13, 14, 28]. Propagation model localization methods analyze the relationship between RSSI values and distances to learn parameters such as the path loss exponent of the propagation path-loss model in the calibration phase. The calibrated propagation model is then applied to convert the signal strength to the estimated distance between transmitter and receiver in the localization phase. In proximity localization methods, an unknown node broadcasts a localization packet to initiate the localization process. Nearby location-known reference nodes then report the RSSI values measured from the packet to a nominated node. The order of reported RSSI values is then used to determine the location of the unknown node. Fingerprinting localization methods measure RSSI values from a set of static nodes during a calibration phase at several locations. The measured RSSI values at a particular location are then used to fingerprint the location. In the localization phase, a node measures RSSI values from the same static nodes and then estimates its location by finding the fingerprinting that is the closest match with the measured RSSI values.

In this paper, based on the concept of integrating AoA Localization and fingerprinting localization for reducing errors, we propose a novel localization scheme, called AoA Localization with RSSI Differences (ALRD). It estimates the AoA for localization in 0.1 s by comparing the RSSI values of beacon signals received from two perpendicularly oriented directional antennas installed at the same place. In the proposed ALRD, we fit RSSI values received from a directional antenna into a parabola function of an AoA between 0° and 90°. We also set up a beacon node with two perpendicularly oriented directional antennas and fit the difference of the signal RSSI values of the two antennas into a linear function of the absolute value of one AoA between 0° and 90°. With the parabola and linear functions, a sensor node can then self-localize itself quickly (within 0.1 s) by observing RSSI values of the beacon signals emitted by the two beacon nodes. The fitting functions can easily be stored in a WSN node, despite their limited storage space, and their inverse functions can be used to speed up the localization process. Hence, ALRD is suitable for mobile sensing and actuating applications in an open and stable environment since it allows a sensor node to fast localize itself with small localization errors. Our experiments demonstrate that the average localization error is 124 cm when deployed in a 10 × 10 m indoor area.

We further propose two methods, namely *maximum-point minimum-diameter* and *maximum-point minimum-rectangle*, to reduce ALRD localization errors by gathering more beacon signals within 1 s for finding the set of estimated locations of maximum density. Such estimated locations are then averaged to obtain the final location estimation. Experimental results obtained demonstrate that the two methods can reduce the average localization error by a factor of about 29%, to 89 cm. Hence, ALRD is suitable for mobile sensing and actuating applications, as it allows a sensor node to quickly localize itself with lower localization errors.

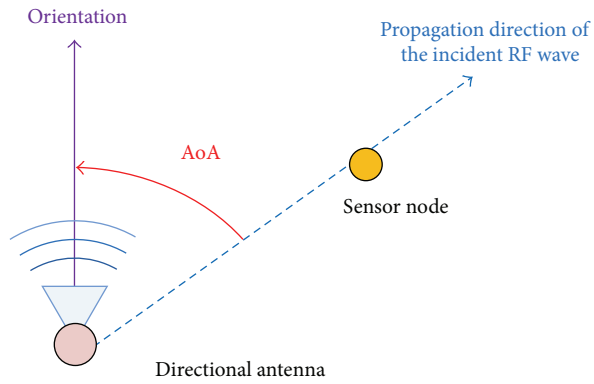


Figure 1: AoA of a sensor node and a directional antenna.

The rest of this paper is organized as follows. We review some AoA determination schemes in Section 2. In Section 3, we describe the proposed localization scheme, ALRD, in detail. Section 4 shows our experimental results. Then, we describe improvements to ALRD and compare it with other schemes in Section 5. Finally, we conclude the paper in Section 6.

2. Related Works

In this section, we review some research that determines AoAs for localization. Amundson et al. developed the RIMA system that uses radio interferometry measurements to estimate the AoA [15]. RIMA estimates the AoA by measuring the TDoA of an interference signal generated by a antenna array. The system consists of a beacon node and a target node. The beacon node is formed by grouping three sensor nodes to form an antenna array. The three sensor nodes are arranged in a manner such that their antennas are mutually orthogonal. Two of the sensor nodes transmit a pure sinusoidal signal at slightly different frequencies to create a low-frequency interference signal. The other sensor node and the target node both measure the phase of the low-frequency signal. The difference in the phase readings measured by these two nodes is then used to estimate the AoA from the beacon node to the target node. Although RIMA can accurately measure the AoA within 1 s, it requires very accurate time synchronization between the beacon node and the target node, which is very difficult to achieve.

Two methods, Estimating Direction-of-Arrival (EDoA) [5] and Rotatable Antenna Localization (RAL) [16], utilize the property of directional antennas to estimate the AoA of a signal. EDoA estimates the AoA of an incoming signal by using a mechanically actuated parabolic reflector. The receiver, which is fixed to a parabolic reflector rotated by a step motor, is used to observe the RSSI values of signals emitted from a transmitter. When the orientation of the reflector is aligned with the direction from the receiver to the

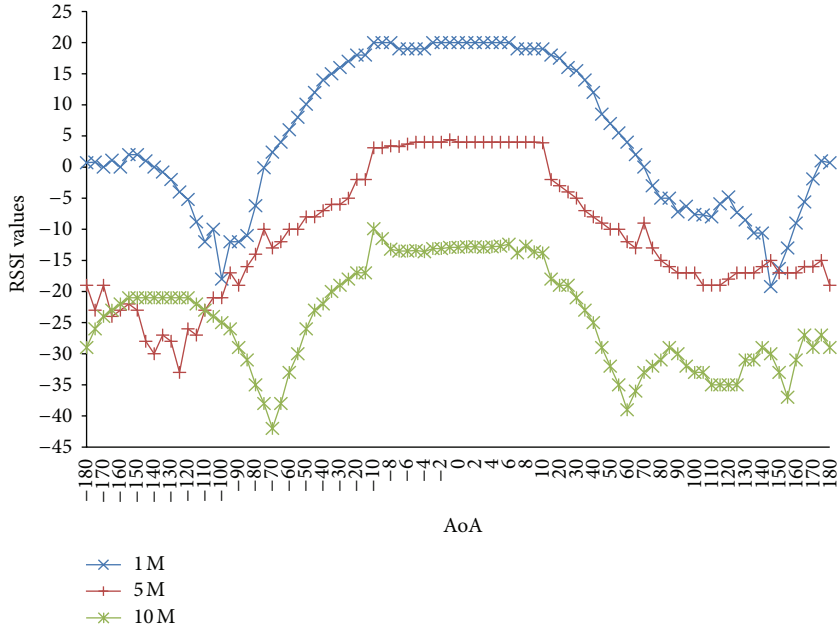


Figure 2: RSSI values of signals received from a directional antenna.

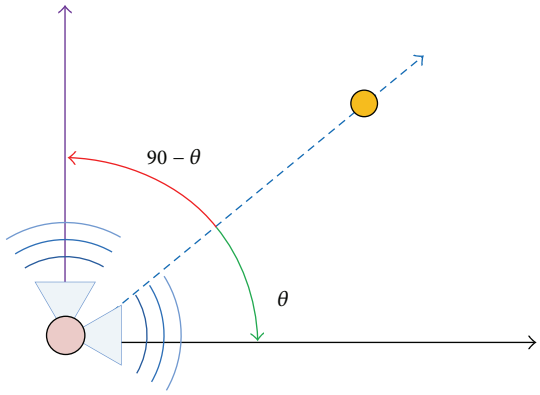


Figure 3: AoAs relative to directional antennas with perpendicular orientations.

transmitter, the receiver will observe the highest RSSI value. Hence, the AoA can be obtained by searching for the reflector orientation in which the highest RSSI value is observed. Their experimental results show that the error in measuring the AoA has a mean of about 4° and a standard deviation of about 8° in both indoor and outdoor environments. However, EDoA needs to take a long time to rotate the reflector for searching the highest RSSI value. In RAL, a beacon node is equipped with a rotatable directional antenna. It regularly rotates its antenna to emit beacon signals in different directions. A sensor node determines the angle from the beacon node to itself by observing the RSSI values of the received beacon signals, which contain the location of the beacon node and the current orientation of its antenna. Similar to EDoA, RAL can determine the AoA by determining the strongest signal. By using the estimated AoAs and locations of two distinct beacon nodes, a sensor node can then calculate its own location with a localization error of 76 cm within a 10 × 10-meter indoor area. Two enhanced methods were further proposed to reduce the localization error by a factor of 10% [16]. EDoA and RAL both need a long time to rotate the antenna or the reflector for observing the variation of the RSSI values while estimating the AoA. Therefore, EDoA and RAL are only suitable for localizing static sensor nodes.

3. The Proposed Scheme

Existing localization schemes using AoAs may take a long time to finish the localization or need very accurate time synchronization. In this section, we propose ALRD that finish localization quickly by learning how the RSSI values vary with the AoA in advance without the need of time synchronization between the anchor node and the target node.

3.1. Preliminary. As shown in Figure 1, we define the AoA as the angle from the propagation direction of an incident RF wave to the orientation of the directional antenna emitting the RF wave. The AoA is positive if it is counterclockwise and negative otherwise. Figure 2 shows a plot of RSSI values over AoA; we observe that if the distance between the sensor node and the directional antenna is fixed, the RSSI varies like a parabolic function of AoA, referenced to the orientation of the directional antenna, between −90° and 90° with a an axis of symmetry at AoA = 0°. Furthermore, we set up two perpendicularly oriented directional antennas installed at the same location (seen in Figure 3). From the results obtained (as shown in Figure 4), we also observe that the difference of the signal RSSI values received by a sensor node, localizing between the orientations of two-directional antennas, varies

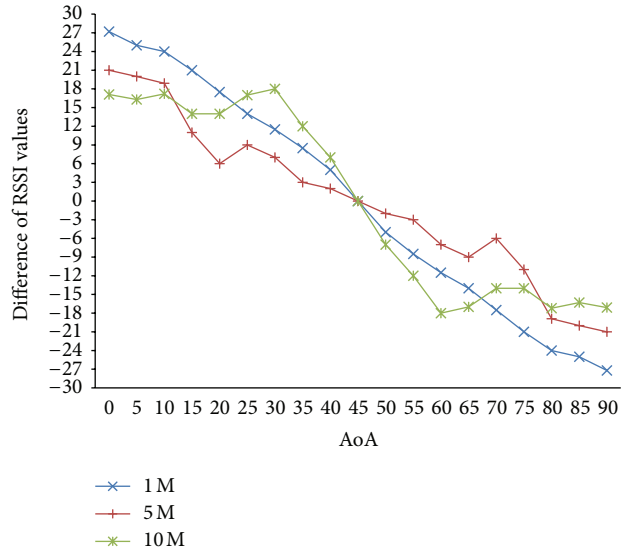


Figure 4: Difference of the signal RSSI values received from two-directional antennas with perpendicular orientations.

like a linear function of the absolute value of the AoA between 0° and 90°. It can be noted that absolute values of AoAs are used and that when one absolute AoA value is θ, the other absolute AoA value is $90 - \theta$. We then take only one absolute AoA value as the representative without ambiguity.

3.2. RSSI Gathering and Analyzing. Before deployment, ALRD needs to gather and analyze RSSI values of the signals that a sensor node receives from a directional antenna at different distances and angles. The measured RSSI values are then analyzed to generate the fitting functions. These fitting functions are then stored into the storage of each sensor node for localization. For better accuracy, the RSSI gathering and analyzing tasks are needed to execute at each new system deployment location, since the environment changes may make the measured RSSI values have some differences. The tasks are described as follows.

(1) Gathering RSSI values: as shown in Figure 5, we set up a directional antenna that can be rotated by an angle θ from the x-axis (or the east direction) (0°) to the y-axis (or the north direction) (90°) and transmits beacon signals containing the rotating angle for every degree. A sensor is placed at the x-axis at a distance of $d, 2d, \ldots, Md$ meters for receiving signals emitted from the antenna for several times (e.g., 100), where d and M are specified values (e.g., $d = 1$ and $M = 10$). The received signal RSSI values are averaged and stored. The gathered RSSI average values are denoted by $G_{kd}(\theta)$, where $k = 1, 2, \ldots, M$, and $\theta = 0°, 1°, \ldots, 90°$.

(2) Performing quadratic regression: for each distance kd, the gathered RSSI values are fitted approximately into a quadratic function $Q_{kd}(\theta)$ of the rotating angle θ, by quadratic regression analysis.

(3) Calculating RSSI differences: for each distance kd, the RSSI difference $D_{kd}(\theta)$ at angle θ is obtained by calculating $G_{kd}(\theta) - G_{kd}(90 - \theta)$. In practice, $D_{kd}(\theta)$ is approximately the difference of RSSI values between two signals that a sensor node receives from two perpendicularly oriented directional antennas installed at the same location.

(4) Performing linear regression: for each distance kd, the RSSI difference $D_{kd}(\theta)$ at angle θ is approximately fitted into a linear function $L_{kd}(\theta)$ by linear regression analysis.

(5) Storing functions: the quadratic and linear approximation functions are loaded into the storage of the sensor nodes before they are deployed.

3.3. ALRD Setup. Figure 6 shows the setup for ALRD. We assume that all sensor nodes are randomly deployed in a planar square area of interest. Two beacon nodes, B_1 and B_2, are deployed in the lower left and lower right corners of the area, respectively. Each beacon node is equipped with two-directional antennas with perpendicular orientations. The antennas of the beacon node in the lower left (or right) corner have either an upright or a horizontal to the right (or left) orientation. The antenna with the upright orientation is called the *vertical antenna*, whereas, the antenna with the left or right orientation is called the *horizontal antenna*. Each beacon node is assumed to know its location and orientations of the two antennas. The beacon nodes transmit beacon signals via the two-directional antennas regularly and alternately. The beacon signal contains the orientation of the antenna and the location of the beacon node, which are expected to reach the whole area of interest.

Note that the setup in Figure 5 can be the basic building block for deploying ALRD in a large indoor localization environment. Figure 7 shows a deployment instance of beacon

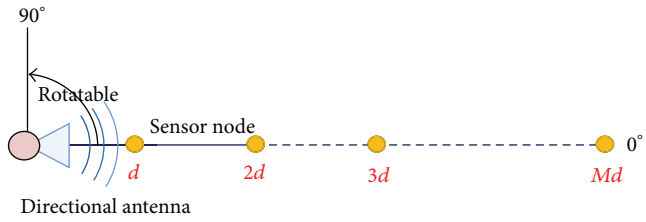


Figure 5: The setup for gathering and analyzing RSSI values.

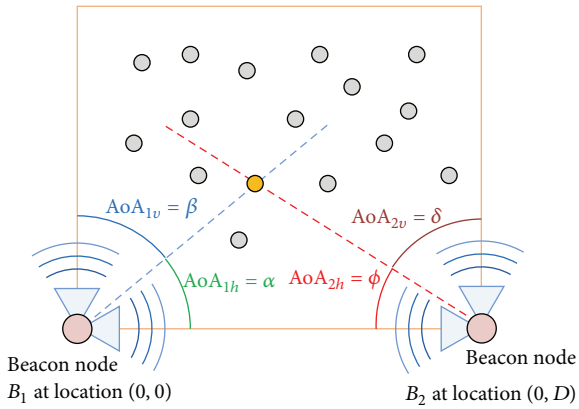


Figure 6: Setup for ALRD.

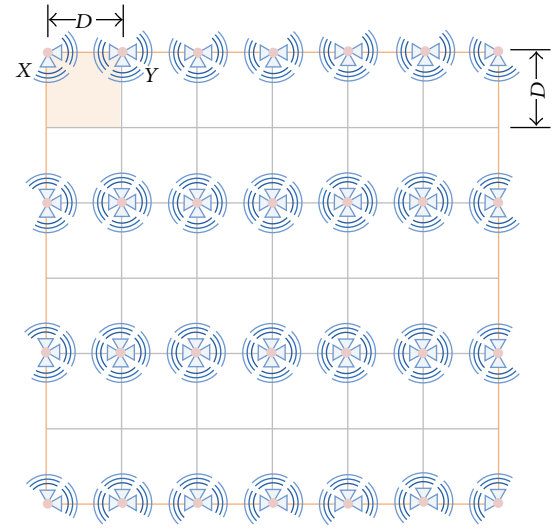


Figure 7: Deployment instance of beacon nodes in a large area.

nodes in a large area. The beacon nodes consist of 1, 2, or 4 pairs of perpendicularly oriented directional antennas. Any two adjacent beacon nodes are D units away from each other in the horizontal direction and $2D$ units away from each other in the vertical direction. Based on the setup of Figure 6, we can see that any two adjacent beacon nodes in the horizontal direction can properly localize target nodes within an area of D by D units. For example, in Figure 7, the beacon nodes X and Y can properly localize target nodes within the shaded area. Hence, ALRD can help localize sensor nodes in a large area by installing a lot of beacon nodes.

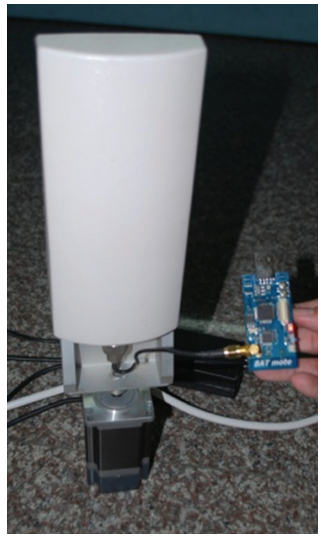
3.4. Localization Procedure. In ALRD, a sensor node executes the following steps to estimate its location.

(1) Receiving beacon signals: in order to localize itself, the sensor node needs to collect the signal RSSI values R_{1h} and R_{1v} of the horizontal and vertical antennas of beacon node B_1. It also needs to collect the signal RSSI values R_{2h} and R_{2v} of the horizontal and vertical antennas of beacon node B_2.

(2) Estimating distance: for each distance kd, for $k = 1, 2, \ldots, M$, two absolute values of AoAs, α_{kd} and β_{kd}, corresponding to the horizontal and vertical antennas of the beacon node B_1 are obtained by calculating $\alpha_{kd} = Q_{kd}^{-1}(R_{1h})$ and $\beta_{kd} = Q_{kd}^{-1}(R_{1v})$, for $0^\circ \le \alpha_{kd}$ and $\beta_{kd} \le 90^\circ$. Here, $Q_{kd}^{-1}(\cdot)$ is an inverse function of the quadratic function $Q_{kd}(\cdot)$ obtained in RSSI gathering and analysis. As shown in Figure 6, $\alpha_{kd} + \beta_{kd}$ should ideally be 90°. Therefore, the sensor node can obtain a rough estimate of the distance from itself to the beacon node B_1, by finding kd for $k = 1, 2, \ldots, M$, such that $\alpha_{kd} + \beta_{kd}$ is closest to 90°. Let the discovered kd be denoted by kd_1. Similarly, by R_{2h} and R_{2v}, the sensor node can find kd such that $\phi_{kd} + \delta_{kd}$ is closest to 90°, where ϕ_{kd} and δ_{kd} are two absolute values of AoAs (where $0^\circ \le \phi_{kd}$ and $\delta_{kd} \le 90^\circ$) corresponding to the horizontal and vertical antennas of the beacon node B_2 and $\phi_{kd} = Q_{kd}^{-1}(R_{2h})$ and $\delta_{kd} = Q_{kd}^{-1}(R_{2v})$. Let the discovered kd be denoted by kd_2.

(3) Estimating AoA: the distance estimate kd_1 obtained in Step 2 is used to choose a proper linear approximation function L_{kd_1} for estimating the AoA of the sensor node corresponding to the beacon node B_1. The AoA corresponding to the horizontal antenna of B_1 is calculated as $\alpha_{kd_1} = L_{kd_1}^{-1}(R_{1h} - R_{1v})$, where $L_{kd_1}^{-1}(\cdot)$ is the inverse function of the linear function $L_{kd_1}(\cdot)$ obtained in the RSSI gathering and analysis stage. Similarly, by kd_2 and $(R_{2h} - R_{2v})$, the AoA corresponding to the horizontal antenna of the beacon node B_2 can be calculated as $\phi_{kd_2} = L_{kd_2}^{-1}(R_{2h} - R_{2v})$.

(4) Calculating location: after obtaining the AoAs α_{kd_1} (or α for short) and ϕ_{kd_2} (or ϕ for short) corresponding to the horizontal antennas of the two beacon nodes



(a)

(b)

Figure 8: (a) The beacon nodes used in RSSI gathering and analysis. (b) The beacon node used in localization.

B_1 and B_2, the sensor node can determine its location (x, y) by calculating

$$(x, y) = \left(\frac{D \times \tan\phi}{\tan\alpha + \tan\phi}, \frac{D \times \tan\alpha \times \tan\phi}{\tan\alpha + \tan\phi} \right), \quad 0 \le x,\ y \le D, \tag{1}$$

where D is the known distance of B_1 and B_2. Note that we can also calculate this by using only kd_1 and kd_2. However, our experiments show that using α_{kd1} and ϕ_{kd_2} results in locations that are more accurate. This explains why ALRD uses R_{1h}, R_{1v}, R_{2h}, and R_{2v} to calculate kd_1 and kd_2, uses kd_1 and kd_2 to calculate α_{kd_1} and ϕ_{kd_2}, and then uses α_{kd_1} and ϕ_{kd_2} to calculate the location.

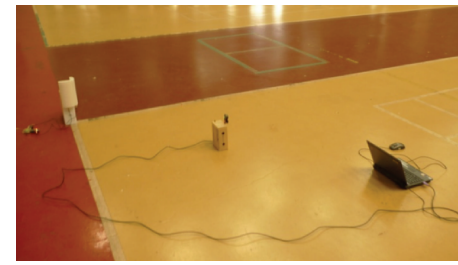


Figure 9: The ALRD experimental setup.

4. Experiment Results

In this section, we describe the implementation of ALRD and the results of the experiments using the implementation.

4.1. Implementation. The sensor nodes and the beacon nodes of the proposed ALRD scheme are implemented in nesC with TinyOS support on the Moteiv BAT mote sensor. The BAT mote sensor has a Texas Instruments MSP430 F1611 microcontroller running at 8 MHz with 10 kB RAM and 48 kB flash memory. It is equipped with the Chipcon CC2420 IEEE 802.15.4 compliant wireless transceiver using the 2.4 GHz band with a 250 kbps data rate. With an integrated onboard omnidirectional antenna, the BAT mote sensor has a maximum transmission range of 50 m (indoor) or 125 m (outdoor).

The beacon node used for RSSI gathering and analysis is also attached with a Maxim AP-12 panel antenna, which is rotated by a Fastech Ezi-Servo 28 L step motor, as shown in Figure 8(a). The beacon node used in localization is attached with two AP-12 panel antennas with perpendicular antennas, as shown in Figure 8(b). Its horizontal and vertical beamwidths are 65° and 28°, respectively.

4.2. Experimental Setup. We installed the ALRD setup in a 10×10 m region of an indoor basketball court for conducting experiments as shown in Figures 9 and 10. Two beacon nodes were set up at two ends of the edge of the experiment area, and the localization accuracy was tested at 81 grid points (as arranged in Figure 10). Since the largest distance between a measurement point and a beacon node is about 12.73 m, we gathered and analyze the RSSI values of signals emitted at distances of $1, 2, \ldots, 13$ m for every degree from 0° to 90°. The RSSI value for each distance and each degree was obtained by averaging 100 measurements.

The gathered RSSI values are shown in Figure 11. The interval for gathering the RSSI values is set as 1 m because the RSSI values will be indistinguishable if the interval is too small. To reduce the gathering time and space used for storing the approximation functions, we only measured RSSI values for one of the four antennas of the same type of the two beacon nodes in our experiments. The coefficients of determination, R^2, of all the approximation functions are shown in Figure 12. We note that the coefficients of determination are high and all exceed 0.96. Therefore, the approximation functions are very suitable for expressing the measured RSSI values and RSSI differences.

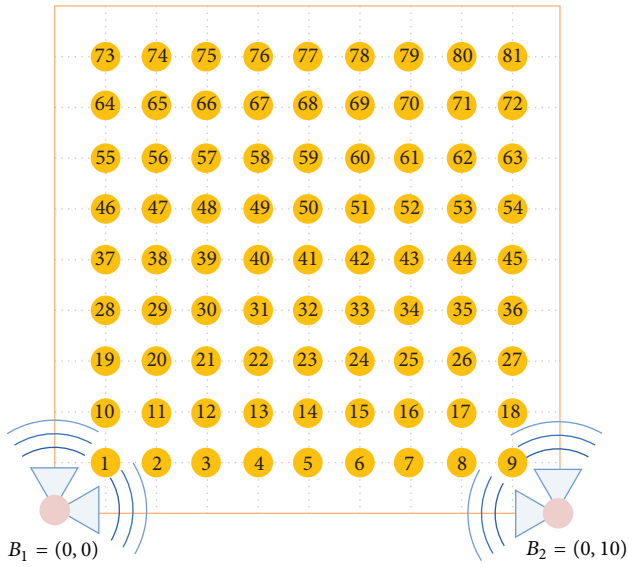


Figure 10: The ALRD experimental setup and 81 grid points for testing.

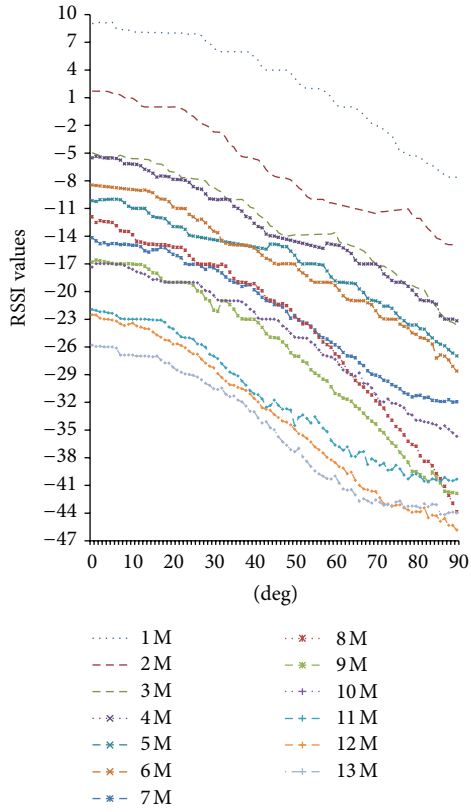


Figure 11: The gathered RSSI values.

4.3. Localization Errors. The localization accuracy is tested at the 81 grid points shown in Figure 10. The beacon nodes transmit beacon signals via each of their antennas 10 times per second. Therefore, a sensor node can localize itself 10 times per second. We take the average of 10 localization results and plot the cumulative distribution in Figure 13. The average

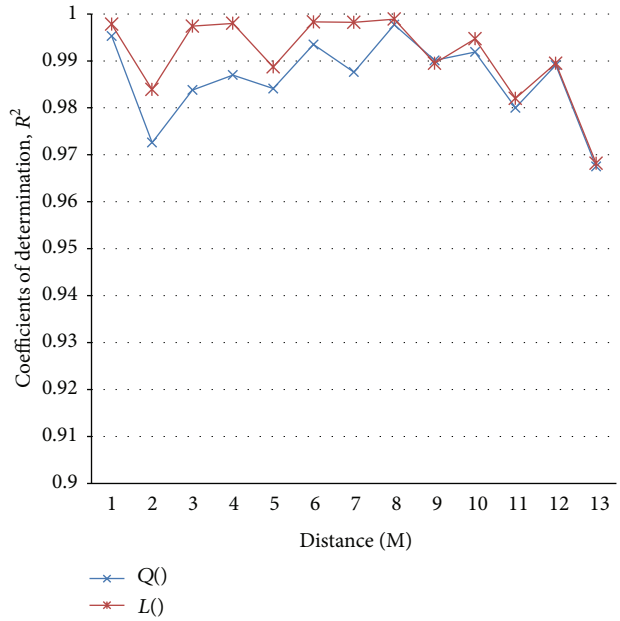


Figure 12: The coefficients of determination, R^2, of the quadratic (Q) and linear (L) approximation functions for different distances.

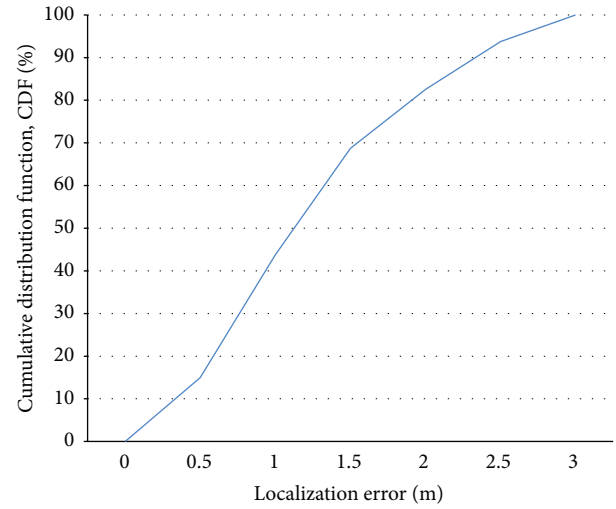


Figure 13: Cumulative distribution function of the localization error.

localization error of the localization experiment is 124 cm. In Figure 14, we use different colors to represent the localization errors of the test points. The brighter color indicates the smaller localization error. As Figure 13 shows, the test points that lie in the middle of the region have smaller localization errors. This can be explained by Figure 4, in which the curves almost look like straight lines in the middle.

5. Improvement

As the results show, ALRD can let sensor nodes localize themselves by measuring RSSI values of signals from two beacon

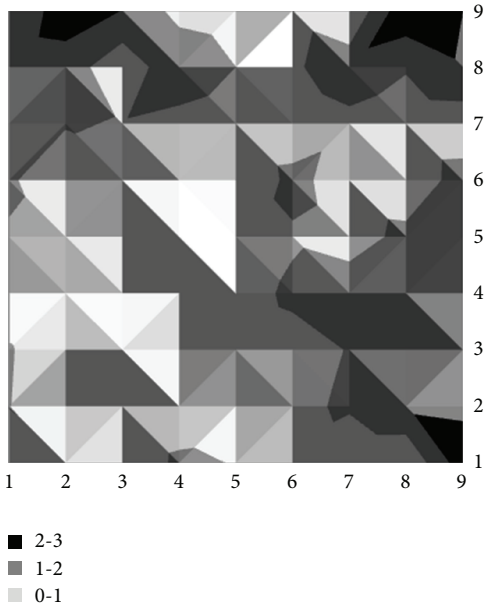


Figure 14: The localization error distribution.

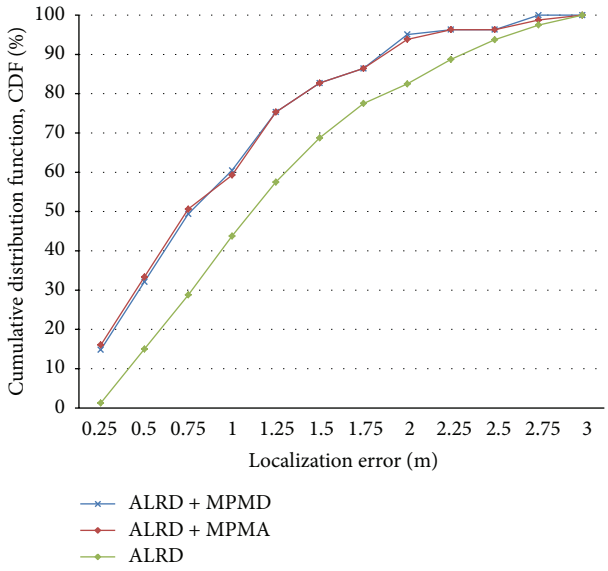


Figure 15: Cumulative distributions of localization errors.

nodes in a short time (0.1 s). However, the measured RSSI values may be influenced by environmental interferences so the estimated location may deviate from the real location and has a localization error. Thus, if a node spends more time measuring more RSSI values, then more location estimations can be made, which in turn can reduce the deviations. Based on the concepts introduced in [28], we propose two methods, namely, maximum-point minimum-diameter (MPMD) and maximum-point minimum-rectangle (MPMR), to remove some estimated location from a set of estimated locations for the purposes of reducing the localization error.

Assuming that P is the set of estimated locations, MPMD and MPMR remove some locations by finding a subset $P' \subseteq P$ with the largest density. The density ρ of a set of estimated locations is defined as follows:

$$\rho = \frac{N}{\text{Dia}}\ (\text{MPMD}) \quad \text{or} \quad \rho = \frac{N}{\text{Area}}\ (\text{MPMR}), \tag{2}$$

where N is the cardinality of the set, Dia is the diameter of the locations in the set, and Area is the area of the smallest axis-parallel rectangle containing all locations in the set. The diameter Dia of a set of locations can be obtained by finding a pair of locations with the longest distance between them. The Area of a set of locations can be obtained by finding the four extremes (i.e., the leftmost, rightmost, highest, and lowest extremes) and calculating the area of the rectangle bounded by them.

The following steps are executed by a sensor node to apply MPMD or MPMR to obtain a subset P' of a set P of locations such that P' has the largest density.

Step 1. If $|P| < 3$, then return P.

Step 2. Derive subset P' of P by removing the location with the maximum summation of the distances from itself to other locations in P, and calculate the density ρ' of P'.

Step 3. If $\rho > \rho'$, then return P; otherwise, set $P = P'$ and go to Step 1.

By using the set of estimated locations with the maximum density returned by MPMD and MPMR, we can then calculate the average of all locations to derive a new estimated location with a small localization error. Table 1 shows the average of the localization errors after applying MPMD and MPMR to 10 localization results collected by a sensor node within 1 second. Figure 15 shows the cumulative distributions of the ALRD localization errors and those improved by MPMD and MPMR. As shown, MPMD is more suitable than MPMR for sensor nodes because it makes similar improvements as MPMR but has less computational overheads.

Table 2 compares ALRD with other three localization schemes, namely, EDoA [4], RAL [13], and RIMA [15]. As we have mentioned previously, EDoA and RAL take a long time to localize sensor nodes because they have to rotate the antennas or the reflector. RIMA is able to accurately localize a target node in a short time, but it requires time synchronization between the beacon node and the target node. By contrast, ALRD can localize sensor nodes in a short time and provides relatively low localization errors without the need for time synchronization.

Table 1: Comparisons of localization errors.

	ALRD	ALRD + MPMD	ALRD + MPMR
Localization errors (cm)	124	90	89
Improvement	—	28%	28.8%

Table 2: Comparison of localization schemes.

Method	#BN	#Antenna	SYN	Time	AE	LE
ALRD	2	2/BN	N	<1 s	6°–8°	89 cm
RIMA	1	3/BN	Y	1.6 s	3°	—
EDoA	—	1/SN	N	200 s	4°–8°	—
RAL	2	2/BN	N	180 s	4°–8°	76 cm

BN: the number of beacon nodes.
Antenna: the number of antennas per beacon node (BN) or per sensor node (SN).
AE: angle error.
LE: localization error.
SYN: time synchronization requirement.

6. Conclusion

In this paper, we proposed AoA Localization with RSSI Differences (ALRD) to estimate angle of arrival (AoA) by comparing the received signal strength indicator (RSSI) values of beacon signals received from two perpendicularly oriented directional antennas installed at the same place. We have implemented and installed ALRD in a 10×10 m indoor environment. Our experimental results showed that a sensor node can estimate its location by using only four beacon signals within 0.1 s with an average localization error of 124 cm. Hence, ALRD conserves the time and energy spent on localization. Furthermore, we proposed two methods, namely, maximum-point minimum-diameter (MPMD) and maximum-point minimum-rectangle (MPMR), to reduce ALRD localization errors by gathering more beacon signals within 1 s to find the set of estimated locations of maximum density. The results demonstrated that MPMD and MPMR can reduce the localization error by a factor of about 29% to 89 cm. Thus, as ALRD allows a sensor node to quickly localize itself with lower errors; it is suitable for mobile sensing and actuating applications.

As our experiments show, it is sufficient to gather RSSI values for only one of four antennas of the same type to achieve sufficient localization accuracy. By equipping antennas of the same type to all beacon nodes, the sensor nodes merely need to store the quadratic and linear approximation functions of one antenna. In the future, we will focus on applying ALRD to realize a large-area localization system. Moreover, we will also try to apply different types of directional antennas and their combinations to ALRD in the hope of further reducing localization error.

References

[1] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System: Theory and Practice*, Springer, Berlin, Germany, 1997.

[2] T. Y. Chen, C. C. Chiu, and T. C. Tu, "Mixing and Combining with AOA and TOA for Enhanced accuracy of mobile location," in *Proceedings of the 5th European Personal Mobile Communications Conference*, pp. 276–280, 2003.

[3] N. Patwari and A. O. Hero III, "Location estimation accuracy in wireless sensor networks," in *Proceedings of the 36th Asilomar Conference on Signals Systems and Computers*, pp. 1523–1527, November 2002.

[4] C. Li and Z. Weihua, "Hybrid TDOA/AOA mobile user location for wideband CDMA cellular systems," *IEEE Transactions on Wireless Communications*, vol. 1, no. 3, pp. 439–447, 2002.

[5] F. Gustafsson and F. Gunnarsson, "Positioning using time-difference of arrival measurements," in *Proceedings of the IEEE International Conference on Accoustics, Speech, and Signal Processing*, pp. 553–556, April 2003.

[6] W. Xiao, Y. Weng, and L. Xie, "Total least squares method for robust source localization in sensor networks using TDOA measurements," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 172902, 8 pages, 2011.

[7] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings of the 19th IEEE International Conference on Computer Commmunications (IFOCOM '00)*, pp. 775–784, 2000.

[8] P. K. Sahoo and I. Hwang, "Collaborative localization algorithms forwireless sensor networks with reduced localization error," *Sensors*, vol. 11, no. 10, pp. 9989–10009, 2011.

[9] J. Shirahama and T. Ohtsuki, "RSS-based localization in environments with different path loss exponent for each link," in *Proceedings of the IEEE Vehicular Technology Conference-Spring, VTC*, pp. 1509–1513, May 2008.

[10] J. Lee, W. Chung, and E. Kim, "A new range-free localization method using quadratic programming," *Computer Communications*, vol. 34, no. 8, pp. 998–1010, 2011.

[11] Z. Liu and J. Chen, "A new sequence-based iterative localization in wireless sensor networks," in *Proceedings of the International Conference on Information Engineering and Computer Science (ICIECS '09)*, December 2009.

[12] K. Yedavalli and B. Krishnamachari, "Sequence-based localization in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 81–94, 2008.

[13] S. Fang and T. Lin, "A dynamic system approach for radio location fingerprinting in wireless local area networks," *IEEE Transactions on Communications*, vol. 58, no. 4, pp. 1020–1025, 2010.

[14] P. M. Scholl, S. Kohlbrecher, V. Sachidananda, and K. Van Laerhoven, "Fast Indoor Radio-Map Building for RSSI-based Localization Systems," in *Proceedings of the 9th International Conference on Networked Sensing System*, pp. 1–2, 2012.

[15] I. Amundson, J. Sallai, X. Koutsoukos, and A. Ledeczi, "Radio interferometric angle of arrival estimation," *Lecture Notes in Computer Science*, vol. 5970, pp. 1–16, 2010.

[16] S. J. Halder and W. Kim, "A fusion approach of RSSI and LQI for indoor localization system using adaptive smoothers," *Journal of Computer Networks and Communications*, Article ID 790374, 10 pages, 2012.

[17] I. Amundson, J. Sallai, X. Koutsoukos, and A. Ledeczi, "Mobile sensor navigation via RF-based angle of arrival localization," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 842107, 15 pages, 2012.

[18] I. Amundson, J. Sallai, X. Koutsoukos, and A. Ledeczi, "Mobile sensor navigation using rapid RF-based angle of arrival localization," in *Proceedings of the 17th IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 316–325, Chicago, Ill, USA, 2011.

[19] A. Cenedese, G. Ortolan, and M. Bertinato, "Low-density wireless sensor networks for localization and tracking in critical environments," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 6, pp. 2951–2962, 2010.

[20] B. N. Hood and P. Barooah, "Estimating DoA from radio-frequency rssi measurements using an actuated reflector," *IEEE Sensors Journal*, vol. 11, no. 2, pp. 413–417, 2011.

[21] J. R. Jiang, C. H. Lin, and Y. J. Hsu, "Localization with Rotatable directional antennas for wireless sensor networks," in *Proceedings of the 39th International Conference on Parallel Processing Workshops (ICPPW '10)*, pp. 542–548, September 2010.

[22] T. Kim, M. Shon, M. Kim, D. S. Kim, and H. Choo1, "Anchor-node-based distributed localization with error correction in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 975147, 14 pages, 2012.

[23] Y. Liu, X. Yi, and Y. He, "A novel centroid localization for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 829253, 8 pages, 2012.

[24] A. Nasipuri and K. Li, "Experimental evaluation of an angle based indoor localization system," in *Proceedings of the 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, pp. 1–9, 2006.

[25] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1, pp. 267–280, 2003.

[26] C. H. Ou, "A localization scheme for wireless sensor networks using mobile anchors with directional antennas," *IEEE Sensors Journal*, vol. 11, no. 7, pp. 1607–1616, 2011.

[27] R. Peng and M. L. Sichitiu, "Angle of arrival localization for wireless sensor networks," in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad hoc Communications and Networks (Secon '06)*, pp. 374–382, September 2006.

[28] R. Atanassov, P. Bose, M. Couture et al., "Algorithms for optimal outlier removal," *Journal of Discrete Algorithms*, vol. 7, no. 2, pp. 239–248, 2009.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 797354, 12 pages
http://dx.doi.org/10.1155/2013/797354

Research Article

Pymote: High Level Python Library for Event-Based Simulation and Evaluation of Distributed Algorithms

Damir Arbula and Kristijan Lenac

Department of Computer Engineering, Faculty of Engineering, University of Rijeka, 51000 Rijeka, Croatia

Correspondence should be addressed to Damir Arbula; damir.arbula@riteh.hr

Received 4 November 2012; Accepted 19 December 2012

Academic Editor: Long Cheng

In recent years we have witnessed strong development and widespread use of powerful wirelessly connected platforms, thus the set of the related problems that need to be solved by distributed algorithms is growing rapidly. Some of them present large obstacles in harnessing the full potential of this new technology, so there is an imminent need for a fast and easy evaluation of new ideas and approaches. Simulation is a fundamental part of distributed algorithm design and evaluation process. In this paper, we present a library for event-based simulation and evaluation of distributed algorithms. This library provides a set of simple but powerful tools with a goal to ease virtual setup of a complex system such as a distributed network of communicating entities and to define, simulate, and analyze its behavior. In order to reduce a huge problem space inherent in such systems, our library is using a high level of abstraction. This is made possible by a strict and complete definition of the distributed computing environment. The library is implemented in Python whose simple and expressive syntax provides a possibility of minimal implementations and a mild learning curve. In addition to executing automated simulations or larger experiments, the library fully supports interactive mode along with a step-by-step execution, which can be a very powerful combination.

1. Introduction

The evaluation of distributed algorithms calls for adequate simulation and comparison with existing state-of-the-art solutions. Although it seems that this task is straightforward, in practice there are a few issues that need to be handled correctly. Existing simulation environments, like OMNeT++, usually require definition of large set of low level parameters (transmitter frequency, communication protocol, etc.) to simulate the behavior of a system as close to implementation as possible. There is certainly nothing wrong with that approach but when a problem is defined in a more generic way (i.e., anchor free localization of wireless sensor nodes with ranging capabilities) selection of some of those parameters is not problem related, thus they are highly arbitrary. This can lead to a simulation of a more specific case than needed.

Here we present a high level Python library for event-based simulation of distributed algorithms in wireless ad hoc networks. The library allows the user to make implementation of their ideas using Python—a popular, easy to learn, full featured, object oriented programming language. Functionalities provided by the library are implemented without additional layer of abstraction, thus harnessing full power of Python's native highly expressive syntax. Using the library, users can quickly and accurately define and simulate their algorithms.

The library particularly focuses on

(1) fast and easy implementation of ideas and approaches at algorithm level without any specification overhead using formally defined distributed computing environment;

(2) support for two different workflows and their seamless combination: (1) *interactive* control and step by step execution of simulation with easy introspection and modification of all objects in the runtime environment, and (2) *fully automated* creation and modification of simulation environment and running multiple experiments in a clean and minimal way using simple Python scripts;

(3) promoting open source reproducible research, thus encouraging its reuse and reevaluation trough comparisons with new ideas based on easily customized criteria.

Since the library is built upon a formally defined distributed computing environment, implementation of specific algorithm is a straightforward process. This process is additionally alleviated by using interactive console and the native Python debugger in which all objects are directly accessible for introspection and modification.

Python is a modular language, so advanced usage or extension of basic functionalities is easily available by writing additional modules and inheriting from the library core classes. These extensions are encouraged through the open source developing workflow.

In the next section, a brief analysis of currently available simulators and libraries is given. In Section 3, we formally define distributed computing environment giving theoretic foundation and basic principles on which this library is based. This is followed by a short discussion on platform selection, description of some implementation details and possible ways of extending the library beyond its current functionalities. Finally, in Section 5, there is an example of definition, simulation, and analysis of one of the popular localization algorithms using the library in interactive and automated workflow.

2. Related Work

A large number of simulators have been proposed in literature in which algorithms for wireless ad hoc networks can be implemented and studied. These simulators have different design goals and largely vary in the level of complexity and included features. They support different hardware and communication layers assumptions, focus on different distributed networks implementations and environments, and come with a different set of tools for modeling, analysis, and visualization. Classical algorithms include NS-2, OMNeT++, J-Sim, TOSSIM, and others.

NS-2 [1] is a discrete event simulator for general network simulation. It is probably the most widely used network simulator for research. NS-2 provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless networks. It was originally targeted to IP networks but extensions [2, 3] for wireless sensor networks have been proposed in the past. NS-2 allows for a detailed simulation tracing and comes with the simulation tool called NAM (network animator) for later playback. Due to a very detailed packet level simulation, the NS-2 simulator is not suitable for simulation of very large networks made of thousands of nodes. NS-2 has many different distributions and extensions and a large number of network simulations have been performed with them. However NS-2 has a steep learning curve and requires considerable effort to repeat the simulations and compare with the obtained results. The core of the simulator and most of the network protocol models are written in C++, while OTcl is used for the definition and configuration of the simulation environment. It is available under an open source license.

Wireless sensor network simulation can be performed using Mannasim framework which extends NS-2 by introducing new modules for design, development, and analysis of different WSN applications. The Mannasim framework provides standardized structures for common sensor, cluster heads, and access point nodes on top of NS-2. In a simulation these three types of nodes run different algorithms which are implemented directly in C++.

Another simulation platform which is widely used in the global scientific community is OMNeT++ [4, 5]. It is an extensible, modular, component-based C++ simulation library and framework, used primarily for building network simulators. OMNeT++ offers extensive simulation library that includes support for input/output, statistics, data collection, graphical presentation of simulation data, random number generators, and data structures. Domain-specific functionality such as support for sensor networks, wireless ad hoc networks, Internet protocols, performance modeling, and so forth are provided by model frameworks and developed as independent projects. Extensions for real-time simulation, network emulation, alternative programming languages, database integration, and other functions exist. For the simulation of wireless sensor networks one popular extension is Castalia [6]. Its development was motivated by the desire to provide a realistic channel/radio modeling. However, for the development of high level algorithm and studying of its behavior Castalia is time consuming because one has to account for all the details related to specific low level modeling.

J-Sim [7] is a general purpose simulator written in Java according to the component-based software paradigm. Components are loosely coupled as each component can be designed, implemented, and tested independently. On the top of the autonomous component architecture, a generalized packet switched network model defines the generic structure of a node and the generic network components, both of which can then be used as base classes to implement protocols across various layers. J-Sim was initially designed for wired network simulation, but a wireless extension exists which proposes an implementation of the IEEE 802.11 MAC together with a set of network and protocol components which facilitates the simulation of wireless networks. It supports real-time process-driven simulation.

TOSSIM [8] is a platform-specific simulation library which simulates TinyOS [9] motes at the bit level. It allows emulation of system components and modeling of different network topologies thus providing a realistic setting for measurement of the communication costs of algorithms. TOSSIM is an open source discrete event simulator which directly compiles code written for TinyOS to an executable file that can be run on standard PC equipment. It can run simulations with a few thousand virtual TinyOS nodes. It ships with the graphical user interface TinyViz that can visualize and interact with running simulations.

For a survey and comparison of these and other simulation platforms the reader can refer to [10–12].

One classification [13] divides simulators into three major categories based on the level of complexity:

(1) algorithm level,

(2) packet level, and

(3) instruction level.

Algorithm level simulators focus on the logic, data structure, and presentation of algorithms. These simulators do not consider detailed communication models and, most commonly, they rely on some form of a graph data structure to illustrate the communication between nodes. Packet level simulators implement the data link and physical layers in a typical OSI network stack. Hence, it is common for this type of simulators to include implementations of 802.11b or newer MAC protocols and radio models that account for propagation, fading, collision, noise, and wave diffraction. Instruction level simulators model the CPU execution at the level of instructions or even cycles. They are often regarded as emulators.

According to this classification, Pymote is an algorithm level simulator. In comparison to the abovementioned widely used simulators, Pymote does not provide packet level and instruction level simulation. Instead, it uses abstract models of the communicating entities and the environment thus enabling researcher to focus on a general principles not influenced by large amount of implementation details. This fact makes it considerably easier to learn and more straightforward to use. An additional benefit of such decision is scaling of the simulation environment in a way that can accommodate large networks. Pymote is focused on the design and evaluation of algorithms while also providing tools for quick definition of different network structures. In the remaining part of this section we describe some of the algorithm level simulators that were proposed in the literature.

AlgoSensim [14] is a framework used to simulate distributed algorithms. It focuses on network specific algorithms like localization, distributed routing, flooding, and so forth. It is written in Java and uses XML files for configuration. The framework was published as open source in 2006 but has remained in alpha release since.

Shawn's [15] primary design goals are to simulate the effect caused by a phenomenon, not the phenomenon itself, to improve scalability, and to support free choice of the implementation model. Instead of performing a complete simulation of the MAC layer including radio propagation properties such as attenuation, collision, fading, and multipath propagation, Shawn simulates the effects of a MAC layer for the application like packet loss, corruption, and delay. In this way, while producing similar effects on the application layer, a performance gain is obtained with a more efficient implementation. This additionally enables Shawn to support large-scale network simulation. Shawn is written in C++.

NetTopo [13] is an integrated framework for simulation and visualization of wireless sensor networks written in Java. Its design is also algorithm-oriented with the goal of rapid prototyping of algorithms; however, it derives its motivation by the need to study applications which can run partially in a simulation environment and partially in a physical wireless sensor network testbed. NetTopo supports the simulation of large scale networks and provides a graphical user interface to drive the simulation.

Sinalgo [16] is another simulation framework for wireless networks written in Java which does not simulate the different layers of the ISO network stack and focuses instead on algorithm layer abstraction. It offers a message passing view of the network and can simulate very large networks. Sinalgo comes with a set of available models for node mobility, connectivity, initial distribution, interference, and transmission which a user can extend with his own if necessary. The simulation is usually started from the available graphical user interface, but for long-lasting well-defined simulations it can also be started in batch mode.

One important difference of these algorithm level simulators with Pymote is the programming language and environment. Pymote leverages Python's strengths like ease of learning and use and faster development than C++ and Java, making the solution well suited for rapid prototyping. Further difference is that Pymote naturally supports both interactive and programmed simulation modes enabling and actually fostering quick, often intertwined definition and simulation phases. By using Python's introspection power, that is, its ability to inspect objects at runtime, determine information about them, and make that information available to the user, Pymote is able to interactively explore and manage all the entities involved in the simulation.

3. Distributed Computing Environment

To design a proper distributed algorithm, the environment in which it performs must be strictly defined. Distributed environment and restrictions to problem space used in specifications for making Pymote library are taken from [17]. Main principles governing algorithm operation in this environment are described below.

Distributed computing environment is composed from a set of *computational entities* $\mathscr{E}$ (in our case wireless nodes) and *messages* they interchange. A node $x \in \mathscr{E}$ has the capability to store data in finite local memory M_x consisting from a number of defined registers. One of them, with special function, is the *status* register that can take values from a finite set of states $\mathcal{S}$. Other parts of the node x are *CPU* and *communication*.

Node behavior is reactive. It acts only when it detects one of two possible events: (1) arrival of message, and (2) spontaneous impulse—usually used in random or defined nodes at algorithm initiation. The action that node performs is a result of its state (in status register) and the event, which in our case is the arrival of message as follows:

$$\text{status} \times \text{event} \longrightarrow \text{action}. \tag{1}$$

Distributed algorithm is defined as a set of rules that associate all possible combinations of states and events with specific actions.

The system has homogenous behavior if all nodes run the same algorithm which simplifies its development and analysis. Since every nonhomogenous behavior can be made homogeneous, as described in [17], all nodes in our library are running the same algorithm.

Table 1: Message structure.

Field name	Data type	Description
Source	Node	Sender node instance
Nexthop	Node	Neighboring node instance that is next hop in path to destination
Destination	Node	Destination node instance
Header	String	Message header defining function and structure of sent data
Data	Dict	Any data

There are three types of actions: storing and processing data, message transmission, and changing the value of status register. All other actions, such as measuring some phenomena or relation between nodes, can be thought of as a combination of message transmission, storing, and processing data.

Message transmission between nodes is defined as a transfer of finite sequence of bits. Simple but generic structure and description of fields in messages used in our library is presented in Table 1.

Every node x can send a message to a set of other nodes $N_{\text{out}}(x) \in \mathscr{E}$ and receive it from set $N_{\text{in}}(x) \in \mathscr{E}$. Inherent property of every node is *local orientation* which means that a node can distinguish between its neighbors by their unique ID, so for example it can send a message to a specific neighbor without sending it to the other neighbors. This is utilized through `destination` field in the message structure.

In practice, sending and receiving messages are complex operations liable to failures and spanning through several communication layers with unknown, possibly infinite duration. When designing distributed algorithms, it is very important to make their performance invariant to communication failures and delays.

Up to this point, all network properties have been defined for some general case. Special properties are called *restrictions* because algorithms designed under assumption of this properties are restricted and cannot be applied to a more general, unrestricted, cases.

Current restrictions in our library are as follows.

Bidirectional Links. For every node x, a set of outgoing and incoming links are identical $N_{\text{out}}(x) = N_{\text{in}}(x)$, for all $x \in \mathscr{E}$ and the nodes that belong to this set are called *neighbors* of x.

Connectivity. In cooperative distributed algorithms it is very important that every node can communicate to all other nodes, directly or through intermediate nodes.

Total Reliability. Every sent message eventually, in unknown but finite time, will be received with its content uncorrupted and without any failure occurring.

These restrictions are strong but sensible. They try to limit problem space to an application layer only and remove unwanted overspecification by focusing on a generic case solutions.

Node's memory content and information held in it represent its *local knowledge*. If at least one node in a set $W \subseteq \mathscr{E}$ has information p, it is defined as *implicit knowledge*. If every node in a set W has the same information then it is *explicit knowledge*. Notion of knowledge is very important since the sole purpose of (distributed) algorithm is improving local knowledge of nodes and implicit and explicit knowledge of the network in general.

4. Implementation

The Pymote library is based on goals described in Section 1 and formalism of the distributed computing environment stated in Section 3. We proceed with discussion on platform selection and brief description of the implementation of basic library functionalities. We conclude with a description of some of directions that can be taken to extend them.

4.1. Why Python? Python language with included libraries and tools was selected after research and analysis of a number of existing platforms and frameworks. The selection of Python allows to fulfil the following requirements:

(i) easy to learn and well documented;

(ii) full featured object oriented language. Partial or restricted solutions (i.e., MATLAB) are bound to have a limit regarding supported object oriented features;

(iii) simple and highly expressive. As such it keeps the code clean and minimal, making its usage straightforward;

(iv) support for interactive mode since this kind of workflow is especially suitable for experimentation and analysis. *IPython* [18] is a Python interactive console that provides all major functionalities needed for interactive scientific computing (Figure 1);

(v) introspection features, namely, easy programmatic inspection of all defined object's properties;

(vi) rich scientific functions library and strong support for scientific computing and calculations. Although Python is a general purpose language, in this field it is very competitive. *NumPy* and *SciPy* [19] are the fundamental packages for scientific computing in Python. They add significant power to the interactive Python session by exposing the user to high-level commands and classes needed for all kinds of scientific data manipulation. In addition to these, for plotting and general graphical representation of data, *matplotlib* is a perfect choice;

(vii) promote open source reproducible research: Python is a platform that already promotes similar ideas and that grows with the community of its users and developers.

After long and exhaustive usage, we have concluded that Python completely fulfills the requirements.

4.2. Core Classes. The implementation philosophy followed DRY (Do not Repeat Yourself) and KISS (Keep It Simple Stupid) principles. DRY principle states our intention to make a library that will not rediscover and rewrite already existing

```
Python 2.7.1 (r271:86832, Nov 27 2010, 18:30:46) [MSC v.1500 32 bit (Intel)]
Type "copyright", "credits" or "license" for more information.

IPython 0.13.1 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

Welcome to pylab, a matplotlib-based Python environment [backend: Qt4Agg].
For more information, type 'help(pylab)'.
In [1]:
```

Figure 1: IPython interactive console.

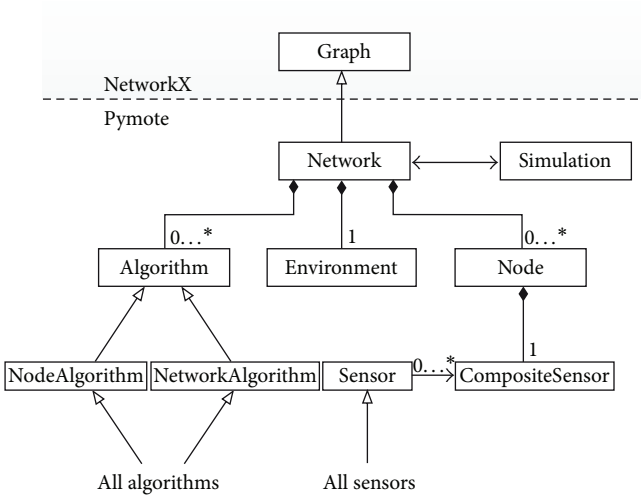


Figure 2: Simplified class diagram.

functionalities but one that will build on top of the existing code. KISS principle reflects the need to simplify library usage and library implementation. To adhere with the DRY principle, Pymote library is placed in such a position that it does not repeat work that has been already done. Since the network is identified as an instance of graph, the only sensible choice was to find a package on top of which this library should be built.

NetworkX [20] is a package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. Its graph definition is as general as possible allowing for easy extension. It has a huge base of graph-related functions and methods already implemented. Pymote library extends from NetworkX and defines few core classes that we describe in this section. Simplified class diagram describing relations between core classes is presented in Figure 2.

Network class is a subclass of the NetworkX's Graph class and as such it implements all methods provided by this class. In addition, all graph-related algorithms implemented in this package are available directly in Pymote. Based on node positions (stored inside network), environment, communication range and channel type, Network is managing creation and removal of links, namely, edges of the underlying Graph. A list of algorithm instances that should be executed is placed inside Network instance. Other responsibilities of Network class include communication or passing messages between neighboring nodes and managing state of algorithms that are executed on a network.

Node class represents a wireless node. Every node has an id to satisfy local orientation property. Additionally, it has a set of memory fields (outbox, inbox, status, and general purpose memory) and communication range property that is used as an argument to a Network class method that is managing communication links. In order to simulate node's knowledge in a more realistic way, a node does not know anything that it would not know in real deployment, for example, the node does not know its position inside the network. If this information should be part of node's knowledge (i.e., node is an anchor), the user could ensure this in experiment setup by equipping the node with an adequate sensor, as demonstrated in Section 5.2.

Instances of *Simulation* class are used to control step-by-step execution of algorithms defined in the network. Simulation is using methods defined in Algorithm class to execute them either on a every node or on a network level, as presented in a sequence diagram in Figure 3. During execution, network and its nodes' states are changing but the simulation instance does not store any data in the process. All data before, during, and after simulation is held inside the network instance which is important as all needed data can be stored simply by serializing and storing the network instance.

Algorithm class represents an executable code that should run inside a network. Currently there are two main subclasses of Algorithm class: NetworkAlgorithm and NodeAlgorithm.

NodeAlgorithm is a classic distributed algorithm that runs in every node. Its start is triggered spontaneously either in randomly chosen nodes or defined ones. Every action after the algorithm initiation is a result of incoming message and state in which the node is in.

There is a notable difference between local termination where the node knows it is done with all actions in the current algorithm and a global termination in which the node knows that all other nodes are also done. Only global termination can be a simple and clear signal to advance further to the next algorithm. In our library the simulation instance (which is used to execute algorithms on a network) is responsible for detecting when a certain algorithm has terminated. It can be detected by the absence of messages (local termination) or by a node returning a clear signal that every node has terminated its execution (global termination).

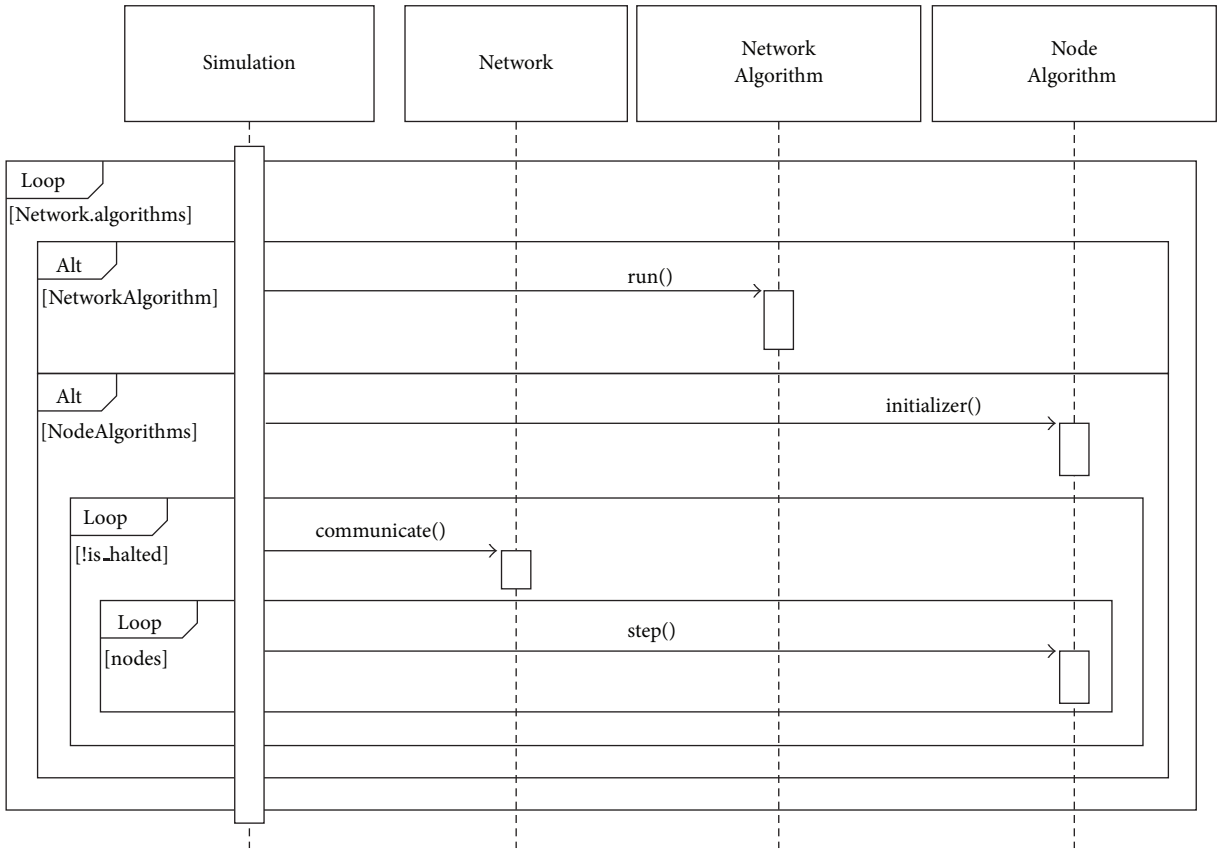


Figure 3: Algorithm sequence diagram.

To simplify the definition and running of distributed algorithms with known behavior, output, and cost, it is allowed for them to use their centralized network version represented by NetworkAlgorithm class. Algorithms that are subclass of NetworkAlgorithm have the ability to improve nodes local knowledge by inserting data directly into their memory. For example, network version of distributed spanning tree algorithm should have the ability to write parent-child relationship directly into every nodes memory. These algorithms are usually helper algorithms that could insert prerequisite knowledge data for algorithms under test. Algorithms under test should always be defined as a proper distributed algorithm or, namely, NodeAlgorithm.

4.3. Extending Base Functionalities. As we have stated in the introduction section, Pymote is primarily focused on the simulation of high level algorithms. Communicating entities should not affect the outcome of the high level algorithm and this is ensured by introducing restrictions such as total reliability, described in Section 3.

Even though they should not affect the outcome, entities and their lower level behavior can affect the algorithm performance, that is, battery usage, reliability of the communication links, and so forth. If the user wants to simulate and inspect lower level behavior or use specific communication protocols it certainly can be done.

For example, low level simulation of battery usage or communication reliability currently can be handled by overriding the `communicate` method of the Network class to simulate and track power consumption or other custom behaviors.

Another example is radio channel simulation which can be done by writing custom ChannelType class and then passing it to network instance. Library is currently implementing two simple ChannelType subclasses: Udg-unit disc graph and SquareDisc in which probability of connection between nodes is equal to $1 - d^2/r^2$ where d is the true distance between nodes and r is the communication range.

The simplest example is the routing protocol which can be implemented as a native NodeAlgorithm with the task of maintaining routing table in the node memory. Data in routing table can be used to appropriately fill `nexthop` field that is already present in the message structure (Table 1) and used in `communicate` method when sending message to nonneighbor node.

All these types of extensions can quickly be incorporated into future versions of the main library. Using collaborative open source workflow, this kind of development is strongly encouraged.

5. Usage

Pymote library supports usage in two distinct workflows. One of the most popular workflows in scientific community

```
class FloodingUpdate (NodeAlgorithm):
   required_params = ('dataKey',)
   default_params = {}

   def initializer (self):
      for node in self.network.nodes():
         if self.initiator_condition(node):
            msg = Message(destination=node,header='initialize')
            self.network.outbox.insert(0,msg)
         node.status = 'FLOODING'

   def flooding(self, node, message):
      if message.header=='initialize':
         node.send(Message(header='Flood',
                           data=self.initiator_data (node)))
      if message.header=='Flood':
         updated_data = self.handle_flood_message (node,message)
         if updated_data:
            node.send (Message(header='Flood',
                               data=updated_data))

   STATUS = {'FLOODING': flooding,}
```

Listing 1: Generic flooding protocol. Function initializer is special function that is issuing spontaneous impulses in the form of *initialize* messages at the beginning of the algorithm execution.

```
class DVHop(FloodingUpdate):

   def initiator_condition(self, node):
      node.memory[self.truePositionKey] = node.compositeSensor.read(node).get('TruePos',None)
      return node.memory[self.truePositionKey] is not None

   def initiator_data(self, node):
      return{node: concatenate((node.memory[self.truePositionKey][:2],[1]))}

   def handle_flood_message(self, node, message):
      if not node.memory.has_key(self.dataKey):
            node.memory[self.dataKey] = {}
      updated_data = {}
      for landmark, landmark_data in message.data.items():
         if landmark==node: continue
         # update only if data is new or new hopcount is smaller
         if not node.memory[self.dataKey].has_key(landmark) or
               landmark_data[2]<node.memory[self.dataKey][landmark][2]:
            node.memory[self.dataKey][landmark] = array(landmark_data)
            # increase hopcount
            landmark_data[2] += 1
            updated_data[landmark] = landmark_data
      # landmarks should recalculate hopsize
      if node.memory[self.truePositionKey] is not None:
         self.recalculate_hopsize(node)
         return updated_data
```

Listing 2: APS DV-hop 1st phase: DVHop. Function *recalculate_hopsize* is omitted. Note that during *initiator_condition* node reads TruePosSensor if it has one.

```
class Trilaterate(FloodingUpdate):

    def initiator_condition(self, node):
        return node.memory[self.truePositionKey] is not None

    def initiator_data(self, node):
        return node.memory[self.hopsizeKey]

    def handle_flood_message(self, node, message):
        if node.memory.has_key(self.hopsizeKey):
            return None
        node.memory[self.hopsizeKey] = message.data
        self.estimate_position(node)
        return node.memory[self.hopsizeKey]
```

Listing 3: APS DV-hop 2nd phase: trilaterate. Function *estimate_position* is omitted.

```
# create network with degree 9
In [1]: netgen = NetworkGenerator(n_min=100,n_max=300,degree=9)
In [2]: net = netgen.generate()
# select landmarks by placing TruePosSensor on them
In [3]: for node in net.nodes()[:10]:
   ....:     node.compositeSensor = CompositeSensor(('TruePosSensor'))
# import algorithms and pass them to network with their parameters
In [4]: from pymote.algorithms.niculescu2003.dvhop import DVHop
In [5]: from pymote.algorithms.niculescu2003.trilaterate import
      Trilaterate
In [6]: net.algorithms = \
   ....:     ((DVHop,
   ....:           {'dataKey': 'dvData',
   ....:            'truePositionKey':'landmarkPos',
   ....:            'hopsizeKey':'hopsize',
   ....:           }),
   ....:      (Trilaterate,
   ....:          {'dataKey': 'dvData',
   ....:           'positionKey':'dvHop',
   ....:           'truePositionKey':'landmarkPos',
   ....:           'hopsizeKey':'hopsize',
   ....:          }),
   ....:     )
In [7]: sim = Simulation(net)
In [8]: sim.run()
INFO     [simulation.py]: Simulation has finished.
# save network with all relevant data on disk
In [9]: write_npickle(net,'net.gz')
```

Listing 4: Interactive session for preliminary algorithm simulations. First we set up a random network in a default environment. Since in the original article presenting the algorithm there is network with 200 randomly distributed nodes and average degree 9 we set up NetworkGenerator with similar settings to get the appropriate instance of Network. After that, algorithms implemented in the last subsection are instantiated and placed in this new network. Since algorithms require some nodes to know their position (landmarks) we fit first 10 nodes with the appropriate sensor using simple for loop. Network is now ready so after instantiation of new Simulation we can run it. At the end by serializing and storing network object all relevant data is preserved to be analyzed later.

```
In [10]: sim.reset() # first reset network algorithm state and nodes memory
In [11]: sim.run(1) # run 1. step of the first algorithm
In [12]: landmark_node = net.nodes()[0]
In [13]: landmark_node.inbox # landmark receiving initializer message
Out[13]:
[
------ Message ------
     source = None
destination = <Node id=1>
     header = 'initialize'
id(message) = 0x908e9f0>]
In [14]: sim.run(1) # run another step
In [15]: landmark_node.outbox # landmark prepared a broadcast message
Out[15]:
[
------ Message ------
     source = <Node id=1>
destination = Broadcasted
     header = 'Flood'
id(message) = 0x90a3d30>]
In [16]: landmark_node.outbox[0].data
Out[16]: {<Node id=1>: array([201.2419, 141.9482, 1.])}
# check if position being sent in message is one being read by TruePosSensor
In [17]: landmark_node.memory['landmarkPos']
Out[17]: array([201.2419, 141.9482])
# another check by directly inspecting network
In [18]: net.pos[node]
Out[18]: array([201.2419, 141.9482])
```

Listing 5: Example of step-by-step simulation and inspection of nodes and network data. In the first step of the DVHop algorithm spontaneous impulses are issued to landmark nodes in the from of special *initialize* messages. As a consequence in the second step landmark node is broadcasting *Flood* message with its true position and hopcount set to 1. After that we check if this is the same data being read by TruePositionSensor and at the end if that really is the landmark's true position by inspecting network data.

```
netgen = NetworkGenerator(degree=9, n_min=100, n_max=300)
For lm_pct in [5, 10, 20, 33]:
  for net_count in range(100):
    net = netgen.generate()
    for node in net.nodes()[:int(lm_pct*len(net.nodes())/100)]:
      node.compositeSensor = CompositeSensor(('TruePosSensor'))
    net.algorithms = ALGORITHMS
    sim = Simulation(net)
    sim.run()
    write_npickle(net, '%d-%d.gz' % (net_count,lm_pct))
```

Listing 6: Simple automated experiment. For each of the defined landmarks percentages (5%, 10%, 20%, 33%) 100 different networks consisting of 100 to 300 nodes with average degree 9 are created. Each of them is fitted with described algorithms which are then executed in a simulation. At the end of each simulation network data is stored in appropriately named compressed file. Definition of variable ALGORITHMS is omitted for brevity.

is using an *interactive console*. This workflow enables free form experimentation, runtime introspection, direct access, and modification of all objects. Another popular workflow is performing a batch of prepared and *automated experiments*.

In this section examples of using the library in both ways are presented. Specifically, we shall try to implement and simulate Ad hoc Positioning System (APS) with DV-hop propagation—a popular algorithm for node localization proposed by Niculescu and Nath [21].

5.1. Algorithm Definition. In Pymote library, algorithms are defined as direct Python implementations. In our example,

```
# inspect non landmark nodes and list their estimated positions
In [19]: for node in net.nodes()[10:]:
    ....:   print node.memory.get('dvHop', 'Notlocalized.')
[466.04   579.44]
[243.25   80.75]
[386.79   69.09]
[254.92   122.66]
[216.36   80.27]
Not localized.
[431.08   102.28]
[140.36   119.35]
etc...
In [20]: estimated = {}
In [21]: for node in net.nodes():
    ....:   if node.memory.has_key('dvHop'):
    ....:     estimated[node] = node.memory['dvHop']
In [22]: get_rms(net.pos,[estimated])
Out[22]: 32.19781563265385
In [23]: show_localized(net,[estimated],show_labels=False)
```

Listing 7: Analyzing nodes memory after simulation. In dvHop key in memory of all successfully localized nodes, there are estimated coordinates data we can inspect. Using library provided, utilities such as `get_rms` function, root mean square error of estimated positions can be quickly estimated. Finally, errors in estimated positions are visualized graphically. Result of the last command is presented on Figure 5.

APS with DV-hop propagation algorithm is based on the following simple idea. Every node in a network should estimate its distance to as many as possible landmark (anchor, beacon) nodes. Landmark nodes are nodes that a priori know their absolute position. Using the estimated distance and a landmark coordinates, the node should be able to estimate its position using trilateration.

The algorithm has two distinct phases as follows.

(1) All landmark nodes are flooding the network with information about their coordinates. As these messages are propagating hop by hop through the network, every node is increasing hopcounter and maintaining minimum received hopcount from each of the landmarks. During this phase landmarks are calculating average hopsize by dividing known distances from other landmarks with respective hopcounts.

(2) After the first phase is done, every landmark makes another (controlled) flooding with information of new estimated hopsize. After the first reception of hopsize from nearest landmark, the node forwards it and drops all other incoming messages. Then it makes distance estimation and finally estimates its position.

These phases are implemented as two separate instances of NodeAlgorithm subclass. Since both phases are using the same flooding protocol to share information we can define a generic algorithm class for flooding. In Listing 1 there is the implementation of base flooding protocol.

In general, NodeAlgorithm is structured as a set of functions that are corresponding to all possible statuses. Each function should be able to handle all types of incoming messages (defined by header) in a sequence of *if* conditions. Since this is a simple protocol with nodes remaining only in one status "FLOODING" and with only one type of message "Flood", the implementation is neat.

Note that in this algorithm function calls (hooks) occur in three places. These are *initiator_condition*, *initiator_data* and *handle_flood_message*. By subclassing this protocol and implementing missing hooks we can make algorithms for both phases, as presented in Listings 2 and 3.

This behavior is in accordance with established DRY principle. Even better, every other algorithm using flooding protocol can reuse this distributed algorithm solely by implementing its own hooks. Thus, object oriented paradigm native to Python can be exploited even in the distributed algorithm implementation.

5.2. Simulation. The best way to initially test and, if needed, debug newly defined algorithms is done by using interactive console. In Listing 4 one example of such interactive session is presented.

Another advantage of running simulation in an interactive console is that on every exception the user is back at the console. There, the user can issue a `%debug` command to enter a debug console and run commands, inspect, and modify all data in every frame of the program stack that raised the exception.

In addition to continuous execution, algorithms can be executed step by step. This is very useful when we want to analyze algorithm in depth and inspect or modify objects during every step of the simulation. Furthermore, it can be handy in a situation when the algorithm does not execute as expected but the error is not explicit enough to raise an exception. There is an example of such execution in Listing 5.

The second type of workflow is an automated experiment. The automated experiment is defined as a simple Python

script that is generating networks, running simulations on them, and eventually storing network data on disk to be analyzed later. Example of one such script is given in Listing 6.

5.3. Analyze Data. Objects can be inspected and data can be analyzed at any moment. For example, during the network setup phase or after the network is loaded from disk it is convenient to visualize network topology. It can be done by calling show method defined on a network object: `net.show()`. Results are displayed in a new window like the one in Figure 4.

Interactive console can be used to analyze experiment data in more detail as presented in Listing 7 and Figure 5, or to return to investigate execution of different parts of specific experiment.

As a complement to console based workflows, Pymote includes a graphical user interface for step-by-step algorithm simulation and visualization. Interface screenshot is presented in Figure 6. Through this interface, users can

(i) save and load network files from disk;

(ii) run and control simulation step by step with immediate visual feedback, such as messages passing, node status, and so forth;

(iii) inspect different objects including network, nodes, and messages simply by clicking on them;

(iv) customize display of network topology, (edges, labels, etc.);

(v) visualize of custom in-node memory structures such as trees;

(vi) pan and zoom control; and

(vii) save visualized data in many different image formats.

6. Conclusion

In this paper, we have presented Pymote, the library that provides support for simulation and analysis of distributed algorithms built on top of comprehensive Python environment. Pymote is designed to allow rapid interactive testing of new algorithms, their analysis and visualization while minimizing developers time. It supports both interactive algorithm simulation and automation of experiments and provides visualization tools for both. It has been deliberately kept simple, easy to use, and extensible.

We plan to continue developing Pymote environment in several different directions as follows:

(i) further development of graphical user interface functionalities for network setup and data analysis,

(ii) implementation of some of the state of the art algorithms from the set of different fields and related problems, and

(iii) adding support for web based version of setup and simulation execution.

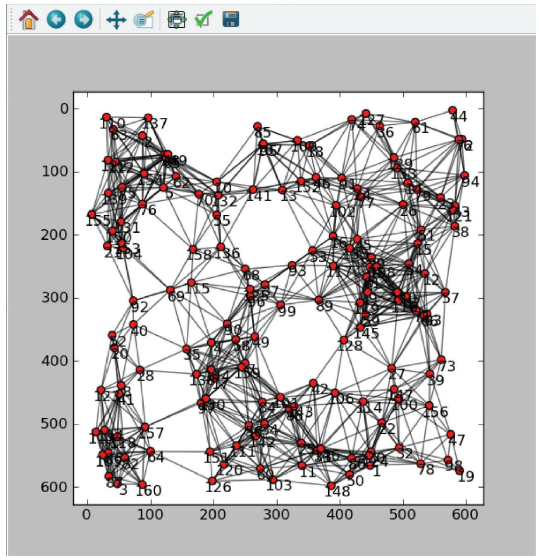


Figure 4: Network topology display.

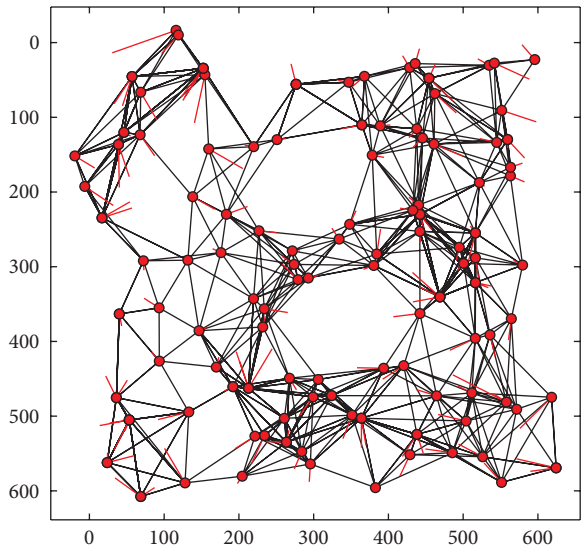


Figure 5: Localization error display.

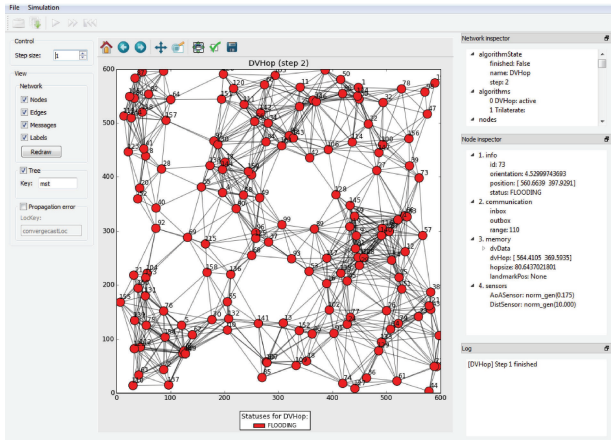


Figure 6: Graphical user interface.

The source code along with the documentation is available as an open source project at https://github.com/darbula/pymote. We hope to have enticed you to take a look at it, use it, or even participate in further development.

References

[1] ns-2, http://www.isi.edu/nsnam/ns/.

[2] I. T. Downard, "Simulating sensor networks in ns-2," Tech. Rep., Naval Research Laboratory, 2004.

[3] Mannasim simulator, http://www.mannasim.dcc.ufmg.br.

[4] Omnet++ simulation system, http://www.omnetpp.org/.

[5] A. Varga, "e omnet++ discrete event simulation system," in *Proceedings of the European Simulation Multi-conference (ESM '01)*, Prague, Czech Republic, June 2001.

[6] A. Boulis, "Castalia: revealing pitfalls in designing distributed algorithms in wsn," in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys '07)*, 2007.

[7] J-Sim, https://sites.google.com/site/jsimofficial/.

[8] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: accurate and scalable simulation of entire TinyOS applications," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, November 2003.

[9] P. Levis, S. Madden, J. Polastre et al., "Tinyos: an operating system for sensor networks," in *Ambient Intelligence*, W. Weber, J. Rabaey, and E. Aarts, Eds., pp. 115–148, Springer, Berlin, Germany, 2005, http://dx.doi.org/10.1007/3-540-27139-2_7.

[10] H. Sundani, H. Li, V. K. Devabhaktuni, M. Alam, and P. Bhattacharya, "Wireless sensor network simulators: a survey and comparisons," *International Journal of Computer Networks*, vol. 2, no. 6, pp. 249–265, 2011.

[11] E. Egea-Lopez, J. Vales-Alonso, A. S. Martinez-Sala, P. Pavon-Marino, and J. Garcia-Haro, "Simulation tools for wireless sensor networks," in *Proceedings of the Summer Simulation Multiconference (SPECTS '05)*, 2005.

[12] S. Mehta, N. Sulatan, and K. S. Kwak, "Network and system simulation tools for next generation networks: a case study," in *Modelling, Simulation and Identification*, A. Mohamed, Ed., InTech, 2010.

[13] L. Shu, M. Hauswirth, H. C. Chao, M. Chen, and Y. Zhang, "Nettopo: a framework of simulation and visualization for wireless sensor networks," *Ad Hoc Networks*, vol. 9, pp. 799–820, 2011.

[14] Algosensim, http://tcs.unige.ch/doku.php/code/algosensim/overview.

[15] A. Kroeller, D. Pfisterer, C. Buschmann, S. Fekete, and S. Fischer, "Shawn: a new approach to simulating wireless sensor networks," in *Proceedings of the Design, Analysis, and Simulation of Distributed Systems (DASD '05)*, San Diego, Calif, USA, 2005, https://github.com/itm/shawn.

[16] Sinalgo, http://dcg.ethz.ch/projects/sinalgo/.

[17] N. Santoro, *Design and Analysis of Distributed Algorithms*, Parallel and Distributed Computing, John Wiley & Sons, 2007.

[18] F. Pérez and B. E. Granger, "Ipython: a system for interactive scientific computing, computing," *Science and Engineering*, vol. 9, no. 3, pp. 21–29, 2007.

[19] E. Jones, T. Oliphant, P. Peterson et al., "SciPy: open source scientific tools for Python," 2001, http://www.scipy.org/.

[20] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using Net-workX," in *Proceedings of the 7th Python inScience Conference (SciPy '08)*, Pasadena, Calif, USA, August 2008.

[21] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Telecommunication Systems*, vol. 22, no. 1–4, pp. 267–280, 2003.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 636052, 7 pages
http://dx.doi.org/10.1155/2013/636052

Research Article

A Novel Human Motion Tracking Approach Based on a Wireless Sensor Network

Sen Zhang,[1] Wendong Xiao,[1] Jun Gong,[2] and Yixin Yin[1]

[1] *School of Automation and Electrical Engineering, University of Science and Technology Beijing, 30 Xueyuan Road, Beijing 100083, China*
[2] *State Key Laboratory of Synthetical Automation for Process Industries, College of Information Science and Engineering, Northeastern University, Shenyang 110819, China*

Correspondence should be addressed to Sen Zhang; zhangsen@ustb.edu.cn

Received 4 October 2012; Accepted 16 November 2012

Academic Editor: Wei Meng

This paper proposes a human motion tracking approach for a daily life surveillance in a distributed wireless sensor network using ultrasonic range sensors. Because the human target often moves with high nonlinearity, the proposed approach applies the unscented Kalman filter (UKF) technique. A novel sensor node selection scheme at each time step considering both the tracking accuracy and the energy cost is presented. Experimental results in a real human motion tracking system show that the proposed approach can perform better tracking accuracy compared to the most recent human motion tracking scheme in the real testbed implementation.

1. Introduction

Human motion tracking is receiving increasing attention from researchers of different fields of study nowadays. The interest is motivated by a wide range of applications, such as wireless healthcare, surveillance, and human-computer interaction. A complete model of human consists of both the movements and the shape of the body. Many of the available systems consider the two modeling processes as separate even if they are very close. In our study, the movement of the body is the target.

Most of the human motion tracking systems are based on vision sensors. Recently, there has been a significant amount of work in tracking people trajectory across multiple image views. Some of the proposed approaches present systems that are capable of segmenting, detecting, and tracking people using multiple synchronized surveillance cameras located far from each other. But they try to hand off image-based tracking from camera to camera without recovering real-world coordinates [1]. Some other work has to deal with large video sequences involved when the image capture time interval is short [2]. The most recent work on vision-based people tracking systems develop wireless sensor networks with low-resolution camera to predict the trajectory of human movement [3]. However, most vision-based approaches to moving object detection are computationally intensive and costly expensive [4]. They often involve intensive real-time computations, such as image matching, background subtraction, and overlapping identification [4]. In fact, in many cases, due to the availability of prior knowledge on target motion kinematics, the intensive and expensive imaging detector array appears inefficient and unnecessary. For example, a video image consisting of 100×100 pixels with 8-bit gray level contains 80 kbits of data, while the position and velocity can be represented by only a few bits [5].

Instead of the centralized processing tracking system based on vision, a promising alternative system named distributed wireless sensor network (WSN) has been quickly developed recently. It consists of many low-cost, spatially dispersed position sensor nodes. Each node can compute and process information that it received and transfer the information among the sensor nodes that are placed within its communication range or to its leader node. Although there are many applications on WSN on target tracking problems [5–10], few papers can be found on human motion tracking in real-time systems [11]. We will develop such a system by WSN in this paper.

From our point of view, human tracking with multiple sensors is an intrinsic multisensor data fusion problem. Multisensor data fusion is such a process through which we combine readings from different sensor nodes, remove inconsistencies, and pull all the information together into one coherent structure. Although some work of multisensor data fusion in WSN has been proposed [6], the tracking accuracy is still limited because of the high nonlinearity property of the human target. In this paper, a UKF filter is employed to estimate the velocity and position of the human trajectory in WSN. UKF filter has the ability to switch between a high process noise (or alternatively, higher order or turn) model in the presence of maneuvers and a low process noise model in the absence of maneuvers. This point gives the UKF filter its advantage over simpler estimators like the Kalman filter and EKF. Compared to the existing work based on EKF [6], the proposed algorithm can give more accurate estimation by using multiple models for human motion in a realtime tracking system developed in this paper.

The layout of the paper is as follows. Section 2 presents the multiple models for human motion tracking. Section 3 presents the UKF estimator for our application. Section 4 proposes the sensor node selection method for our framework. Section 5 presents the simulation results and experimental results. Conclusions and future work are drawn in Section 6.

2. Problem Formulation

In this section, we formulate the human motion tracking as a distributed multisensor data fusion problem. We consider the human moving in a 2D Cartesian coordinate system. The target state includes the human velocity, the human position in the coordinate, and the turn rate when the trajectory is along a curve. We can build up the system models as follows.

2.1. Coordinated Turn Model. In order to describe the human's more complex trajectory, such as turn left or turn right, here we apply the coordinated turn model similar to that in [5]:

$$\mathbf{x}(k+1) = \mathbf{F}(\mathbf{x}(k)) + \mathbf{G}\mathbf{v}(k),$$

$$\text{where } \mathbf{x}(k+1) = \left[P_x(k+1)\, V_x(k+1)\, P_y(k+1)\, V_y(k+1)\, \omega\right]^T,$$

$$\mathbf{F}(\mathbf{x}(k)) = \begin{bmatrix} P_x(k) + \dfrac{\sin \omega T}{\omega} \cdot V_x(k) - \dfrac{1-\cos \omega T}{\omega} \cdot V_y(k) \\ \cos \omega T \cdot V_x(k) - \sin \omega T \cdot V_y(k) \\ P_y(k) + \dfrac{1-\cos \omega T}{\omega} \cdot V_x(k) + \dfrac{\sin \omega T}{\omega} \cdot V_y(k) \\ \sin \omega T \cdot V_x(k) + \cos \omega T \cdot V_y(k) \\ \omega \end{bmatrix},$$

$$\mathbf{G}_2(K) = \begin{bmatrix} \frac{1}{2}T^2 & T & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{2}T^2 & T & 0 \\ 0 & 0 & 0 & 0 & T \end{bmatrix}^T, \tag{1}$$

where ω is the assumed unknown constant turn rate and $\mathbf{v}(k)$ is the process noise. Although the actual turn rate is not exactly a constant, we can assume that it is not changed in a very short-time interval. For convenience, we assume that $\mathbf{v}$ is a zero mean Gaussian white noise with variance $\mathbf{Q}(k)$.

2.2. System Observation Model. In order to build up the estimation scheme using UKF, the sensor observation model is needed. If sensor j is used, $Z_j(k)$ is applied to denote the kth measurement of the target at time step t_k. The measurement model is given by

$$Z_j(k) = h_j(\mathbf{x}(k)) + v_j(k), \tag{2}$$

where h_j is a (generally nonlinear) measurement function depending on sensor j's measurement characteristic and parameters (e.g., its location). $v_j(k)$ is a variable representing measurement noise in sensor j. It is independent and assumed to be zero-mean Gaussian distribution white noise. The covariance of $v_j(k)$ is $R_j(k)$.

3. UKF Filter-Based Human Tracking

Based on the above coordinated constant turn model and the system observation model, the unscented Kalman filter is applied to estimate the system state variable which includes the target's position coordinate and velocity.

Given the estimate $\hat{\mathbf{x}}(k \mid k)$ of $\mathbf{x}(k)$ and its estimation error covariance $\mathbf{P}(k \mid k)$, in order to avoid the linearization involved in the EKF, the UKF works by generating a set of points whose sample mean and sample covariance are $\hat{\mathbf{x}}(k \mid k)$ and covariance $\mathbf{P}(k \mid k)$, respectively. The nonlinear function is applied to each of these points in turn to yield a transformed sample, and the predicted mean and covariance are calculated from the transformed samples. The samples are deterministically chosen so that they capture specific information about the Gaussian distribution.

For highly nonlinear systems, the UKF has advantages over the EKF. It avoids the linearization that causes substantial errors in the EKF for nonlinear systems and possible singular points in Jacobian matrices. The basic UKF algorithm (one cycle) can be seen in [12]. The following is the details of UKF.

3.1. Form Weighted Samples. The n-dimensional random variable $\mathbf{x}(k)$ with mean $\hat{\mathbf{x}}(k \mid k)$ and covariance $\mathbf{P}(k \mid k)$ is approximated by $2n+1$ weighted samples or sigma points selected by the algorithm

$$\begin{aligned} \chi_0(k \mid k) &= \hat{\mathbf{x}}(k \mid k), \\ W_0 &= \frac{\kappa}{(n+\kappa)}, \\ \chi_i(k \mid k) &= \hat{\mathbf{x}}(k \mid k) + \left(\sqrt{(n+k)\,\mathbf{P}(k \mid k)}\right)_i, \\ W_i &= \frac{1}{\{2(n+\kappa)\}}, \\ \chi_{i+n}(k \mid k) &= \hat{\mathbf{x}}(k \mid k) - \left(\sqrt{(n+k)\,\mathbf{P}(k \mid k)}\right)_i, \\ W_{i+n} &= \frac{1}{\{2(n+\kappa)\}}, \end{aligned} \tag{3}$$

where $\kappa \in R$, $(\sqrt{(n+k)\mathbf{P}(k \mid k)})_i$ is the ith row or column of the matrix square root of $(n+k)\mathbf{P}(k \mid k)$, and W_i is the weight that is associated with the ith point. In theory, κ can be any number (positive or negative) providing that $(n+\kappa) \neq 0$.

3.2. Prediction. Given the set of samples generated by (3), the prediction procedure is as follows.

(a) Each sigma point is instantiated through the process model to yield a set of transformed samples

$$\chi_i (k+1 \mid k) = \mathbf{f}_2 \left[\chi_i (k \mid k)\right]. \tag{4}$$

(b) The predicted mean is computed as

$$\widehat{\mathbf{x}} (k+1 \mid k) = \sum_{i=0}^{2n} W_i \chi_i (k+1 \mid k). \tag{5}$$

(c) The predicted covariance is computed as

$$\begin{aligned} \mathbf{P} (k+1 \mid k) = {} & \sum_{i=0}^{2n} W_i \left\{\chi_i (k+1 \mid k) - \widehat{\mathbf{x}} (k+1 \mid k)\right\} \\ & \times \left\{\chi_i (k+1 \mid k) - \widehat{\mathbf{x}} (k+1 \mid k)\right\}^T. \end{aligned} \tag{6}$$

It is also clear that the predicted measurement is simply:

$$\widehat{\mathbf{z}} (k+1) = \mathbf{H}_2 \widehat{\mathbf{x}} (k+1 \mid k). \tag{7}$$

The difference between the measurement and the predicted observation, named the innovation, can be written as

$$\nu (k+1) = \mathbf{z} (k+1) - \mathbf{H}_2 \widehat{\mathbf{x}} (k+1 \mid k). \tag{8}$$

The covariance of this quantity is

$$\mathbf{s}_\nu (k+1 \mid k) = \mathbf{H}_2 \mathbf{P} (k+1 \mid k) \mathbf{H}_2^T + \sigma_r^2. \tag{9}$$

3.3. Calculate the Kalman Filter Gain. Use the following equation

$$\mathbf{K} (k+1) = \mathbf{P} (k+1 \mid k) \mathbf{H}_2^T \mathbf{s}_\nu^{-1} (k+1 \mid k). \tag{10}$$

3.4. Update. We update the estimation using the following equations:

$$\begin{aligned} \widehat{\mathbf{x}} (k+1 \mid k+1) &= \widehat{\mathbf{x}} (k+1 \mid k) + \mathbf{K} (k+1) \nu (k+1), \\ \mathbf{P} (k+1 \mid k+1) &= \mathbf{P} (k+1 \mid k) - \mathbf{K} (k+1) \mathbf{s}_\nu (k+1) \mathbf{K}^T (k+1). \end{aligned} \tag{11}$$

4. Sensor Node Selection

In this section, the sensor selection method under the UKF filter will be presented. It is assumed that each sensor is able to detect the target and determine its range, and the locations of all the sensors are known. One of the approaches simply selects the nodes closest to the predicted target location as estimated by the tracker [13]. The drawback of the "closest" node approach is that it only roughly selects the sensor nodes and does not consider its contribution to the tracking accuracy and the energy consumption quantitatively and simultaneously. In this paper, we propose an adaptive sensor selection scheme similar to [14] but under UKF filter framework. It jointly selects the next tasking sensor and determines the sampling interval at the same time based on both of the prediction of the tracking accuracy and tracking energy cost.

4.1. Tracking Accuracy. Various measures can be defined based on the state estimation to stand for the tracking accuracy, such as the trace and the determinant of the covariance matrix, Fisher information defined on the Fisher information matrix which is the inverse of the state estimation covariance, eigenvalues of the difference between the desired and the predicted covariance matrix of the state, and entropy of the state estimation distribution. In this paper, based on the constant velocity model and the angular coordinated turn model, the tracking accuracy is reflected by tracking error $\phi(k)$ at time step k which is defined as the trace of the covariance matrix $P(k \mid k)$, that is,

$$\phi (k) = \operatorname{trace} (P (k \mid k)). \tag{12}$$

Given a predefined threshold $\phi_0(k)$, the tracking accuracy at time step k is considered to be satisfactory if

$$\phi (k) < \phi_0 (k), \tag{13}$$

otherwise it is considered to be unsatisfactory.

4.2. Energy Model. Energy consumption is used as the tracking cost. We consider the following energy model. If current sensor i selects sensor j as the next tasking sensor, then the total energy consumed by sensor i in transmission is

$$E_t (i, j) = \left(e_t + e_d r_{ij}^\alpha\right) b_c, \tag{14}$$

where e_t and e_d are decided by the specifications of the transceivers used by the nodes, r_{ij} is the distance between sensor i and sensor j, b_c is the number of bits sent, and α depends on the channel characteristics and is assumed to be time invariant. Energy consumed in receiving is

$$E_r (j) = e_r b_c, \tag{15}$$

where e_r is decided by the specification of the receiver of sensor j. The energy spent in sensing/processing data of b_s bits by sensor j is

$$E_s (j) = e_s b_s. \tag{16}$$

Therefore, the total energy consumption is

$$E (i, j) = E_t (i, j) + E_r (j) + E_s (j). \tag{17}$$

4.3. Adaptive Sensor Selection Scheme. Suppose that the current time step is k and the current tasking sensor is the sensor i which receives state estimation $\hat{x}(k-1 \mid k-1)$ and

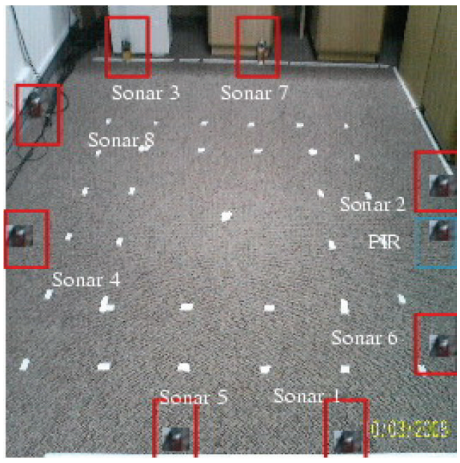


Figure 1: The testbed.

estimation covariance matrix $P(k-1 \mid k-1)$ of the time step $k-1$ from its parent tasking sensor. It first updates the state estimation by incorporating its new measurement $Z_j(k)$ using the UKF algorithm described in Section 2. Then, it uses the sensor scheduling algorithm to select the next tasking sensor j and the next sampling interval Δt_k such that the sensor j can undertake the sensing task at the time $t_{k+1} = t_k + \Delta t_k$. We suppose that Δt_k should be in the range $[T_{\min}, T_{\max}]$, where $T_{\min}$ and $T_{\max}$ are the minimal and maximal sampling intervals, respectively. If sensor j is selected with the sampling interval Δt_k, its associated predicted objective function is defined as

$$J\left(j, \Delta t_k\right) = w\Phi_j\left(k\right) + (1-w)\frac{E\left(i,j\right)}{\Delta t_k}, \tag{18}$$

where $\Phi_j(k)$ is the predicted tracking accuracy according to the UKF algorithm, $E(i,j)$ is the corresponding predicted cost given by (17), the averaged energy consumption over the period. $w \in [0,1]$ is the weighting parameter used to balance the tracking accuracy and the energy consumption.

The sensors are scheduled in the following two tracking methods.

(1) After prediction, none of the sensors can achieve the satisfactory tracking accuracy using any sampling interval in $T_{\min}$ and $T_{\max}$. In this case, Δt_k is set to the minimal sampling interval $T_{\min}$, and the sensor is selected by

$$j^* = \underset{j \subset A}{\arg\min}\left\{J\left(j, T_{\min}\right)\right\}, \tag{19}$$

where A is the candidate sensors that can be selected by sensor i. Generally in (19), $w \neq 0$. The purpose of this mode is to drive the tracking accuracy to be satisfactory as soon as possible with consideration of the energy consumption.

(2) After prediction, at least one sensor can achieve the satisfactory tracking accuracy. In this case, the optimal $(j^*, \Delta t_k^*)$ is selected by

$$\left(j^*, \Delta t_k^*\right) = \underset{j \subset A^*, \Phi(j,k) \leq \Phi_0}{\arg\min}\left\{\frac{E\left(i,j\right)}{\Delta t_k}\right\}, \tag{20}$$

where A^* is the set of sensors that can achieve the satisfactory tracking accuracy. Equation (20) utilizes the objective function in (18) with $w = 0$. The basic idea of this mode is that when the predicted tracking accuracy is satisfactory, the sensors and the sampling interval are selected according to the energy efficiency.

Figure 2: The MicaZ mote.

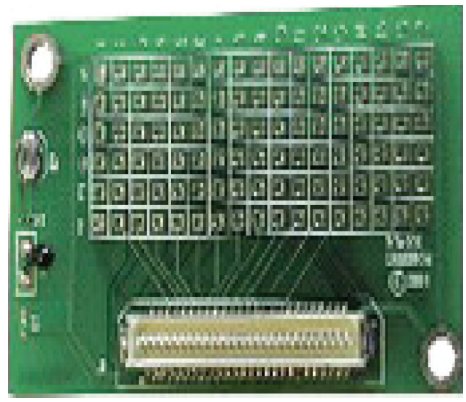


Figure 3: The MDA100CA sensor board.

For simplification, we suppose that the sampling interval is selected from predefined N values $\{T_t\}_1^N$, where $T_1 = T_{\min}$, $T_N = T_{\max}$, and $T_{t_1} < T_{t_2}$ if $t_1 < t_2$. In addition, the set $\{T_t\}_1^N$ is selected such that its values can evenly divide the interval $[T_{\min}, T_{\max}]$ into $N-1$ subintervals.

5. Experimental Results

Our testbed is shown in Figure 1. All the hardwares in the testbed are supplied by Crossbow Technology. The testbed consists of the following hardwares: MicaZ (processor with on-board ZigBee radio), MDA100CA, MIB510 (USB programmer), and SRF02 (active ultrasonic sensor with I2C bus).

Figure 2 shows the MicaZ mote, which operates from the 2400 MHz to 2483.5 MHz band, and uses the Chipcon CC2420, IEEE 802.15.4 compliant, and ZigBee ready radio frequency transceiver integrated with an Atmega128L microcontroller. It has an integrated radio communication transceiver working at 2.4 GHz frequency with a transmission data rate of 250 Kbps and indoor transmission range of 20 to 30 meters. It runs TinyOS and is programmed on nesC.

The MDA100CA series sensor boards have a precision thermistor, a light sensor/photocell, and from general prototyping area. The prototyping area supports connection to all eight channels of the mote's analog to digital converter (ADC 0 to 7), both USART serial ports, and the I2C digital communications bus. The prototyping area also has 45 unconnected holes that are used for breadboard of circuitry. See Figure 3.

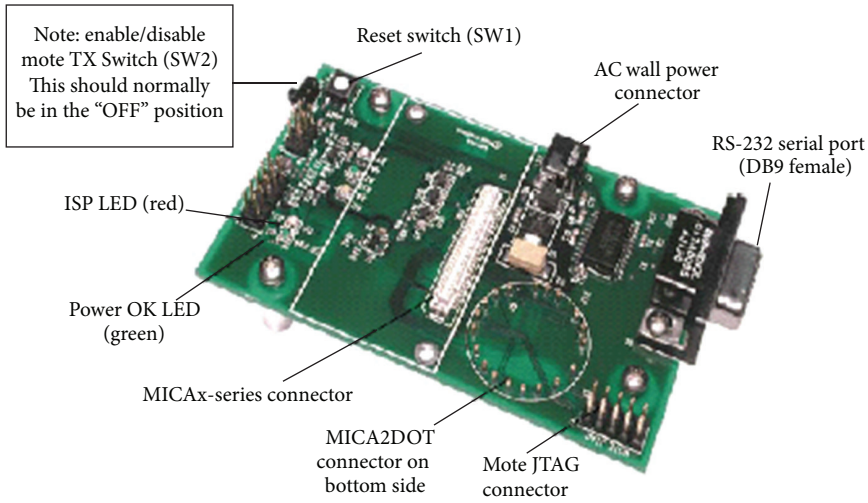


Figure 4: MIB510 programmer board.

Figure 5: The SRF02 ultrasonic sensor.

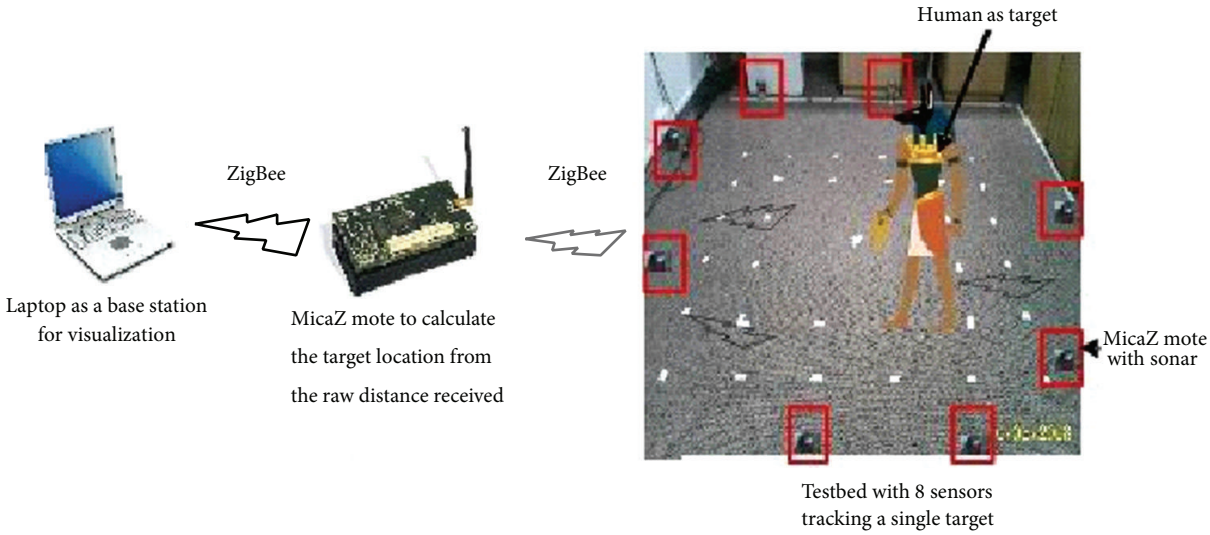


Figure 6: The human tracking system.

The MIB510 interface board (see Figure 4) is a multipurpose interface board used with the MicaZ. It supplies power to the devices through an external power adapter option, and provides an interface for a RS-232 mote serial port and reprogramming port. The MIB510 has an on-board in-system processor (ISP) to program the motes. Code is downloaded to the ISP through the RS-232 serial port. The ISP programs the code into the mote. The ISP and the mote share the same serial port. The ISP runs at a fixed baud rate of 115.2 kbaud. The ISP continually monitors incoming serial packets for a special multibyte pattern. Once this pattern is detected, it disables the mote's serial RX and TX (two legs), then takes control of the serial port.

The SRF02 (see Figure 5) is a single transducer ultrasonic range sensor. It features both I2C and a serial interfaces. I2C interface is used in this project. We use only 8 sensors in

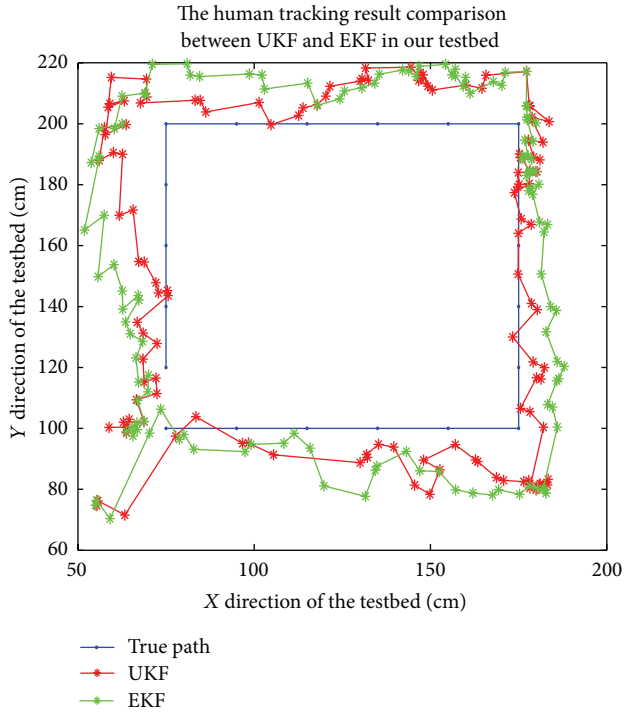


FIGURE 7: The human tracking system experimental result with UKF and EKF.

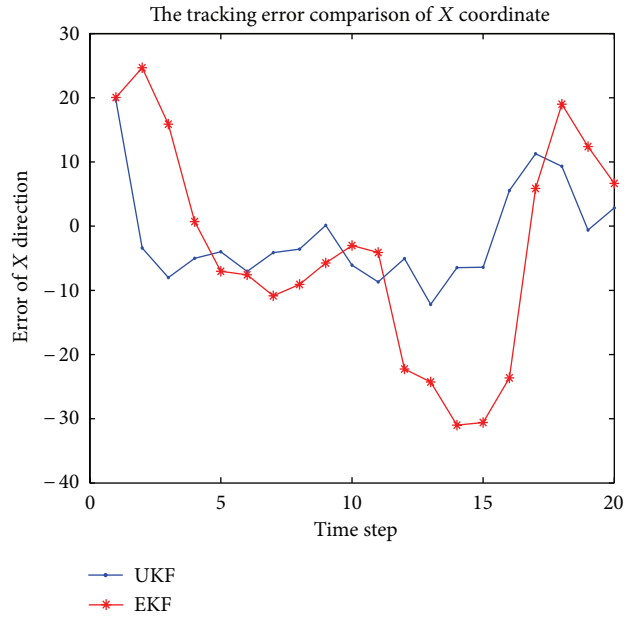


FIGURE 8: The human tracking error comparison in X direction of the testbed with UKF and EKF.

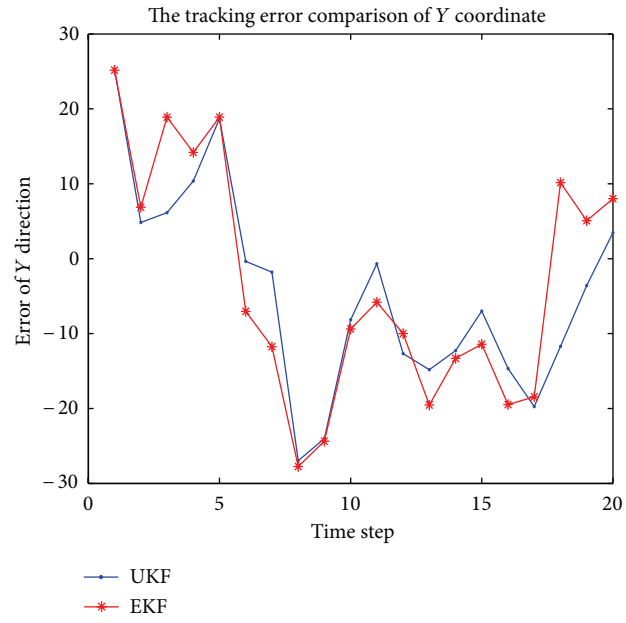


FIGURE 9: The human tracking error comparison in Y direction of the testbed with UKF and EKF.

the testbed. New commands in the SRF02 include the ability to send an ultrasonic burst on its own without a reception cycle and the ability to perform a reception cycle without the preceding burst. SRF02's minimum measurement range is around 15 cm (6 inches). This sensor has a detection angle of 15 degrees and a maximum range of 6 m.

The developed target tracking system, see Figures 1 and 6, is made up of 8 ultrasonic sensor nodes. These 8 ultrasonic sensors arelocated along the edge of the area, respectively, with coordinates (200, 0), (250, 170), (50, 300), (0, 110), (100, 0), (250, 60), (150, 300), and (0, 230). The orientations of the sensors (clockwise from the positive x-axis) are, respectively, 65°, 90°, 50°, 75°, 100°, 110°, 90°, and 120° such that the sound waves are not reflected from nearby walls/obstacles. Each node is allocated with an ID number and an XY coordinate. Their locations are shown in Figure 1 to cover a monitoring area of 2.5 m × 3.0 m. The tracking target is a human. A MicaZ mote will be attached to each sensor node.

On the base station, a laptop is connected to the network through a MicaZ mote for receiving data packets via USB connection. Figure 6 shows the tracking system deployed in the testbed. Upon receiving an initial time synchronizing beacon from processing mote, all sensor nodes will initialize their starting time for sensor nodes. These sensor nodes will broadcast their sensor readings with one sensor reading at a time to the processing mote to avoid sensors' interference. The processing mote will also program the default measurement for each sensor.

The real-time data is collected from a human who is moving around within the sensor coverage area of the testbed. The sensor selection scheme is performed during the data collection process. In order to simplify the sensor selection algorithm, we selected one sensor at each time step in the experiments. The data collected is run by UKF filter-based tracking approach proposed in Section 2, and we compared the experimental results with the method in [13]. Figure 7 to Figure 10 shows the comparison of the result with UKF and EKF, respectively. We can see that the result from UKF is better than the results from EKF in the real testbed.

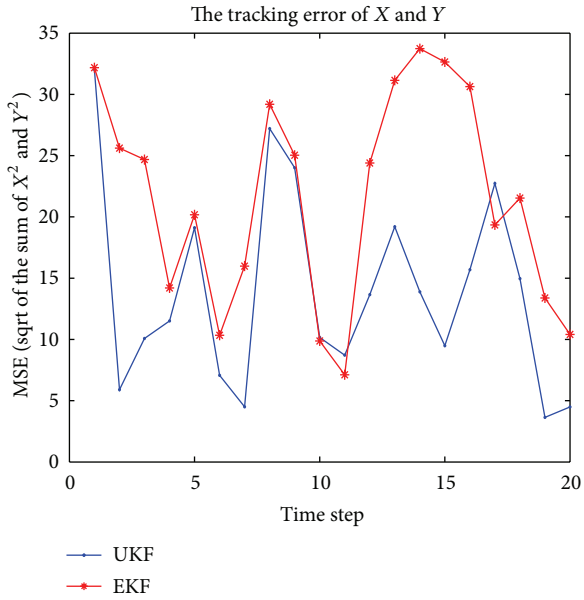


FIGURE 10: The human tracking error comparison of the MSE (mean square root error) of the testbed with UKF and EKF.

6. Conclusions

This paper presents a UKF filter-based adaptive sensor scheduling scheme for human tracking in wireless sensor networks. It uses cheap range sensor nodes in wireless sensor networks by jointly selecting the next tasking sensor and determining the sampling interval based on predicted tracking accuracy and tracking cost under the UKF filter frame. Simulation results show that the new scheme can achieve significant energy efficiency without degrading the tracking accuracy. There are still many issues remaining for future study. Multistep, multisensor selection based adaptive sensor scheduling and sensor scheduling for multitarget tracking are both challenging problems for further investigations.

References

[1] A. Mittal and L. S. Davis, "M2Tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo," in *Proceedings of European Conference on Computer Vision*, pp. 18–36, Copenhagen, Denmark, June 2002.

[2] Q. Cai and J. K. Aggarwal, "Tracking human motion in structured environments using a distributed-camera system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 11, pp. 1241–1247, 1999.

[3] W. Zajdel, A. T. Cemgil, and B. J. A. Brose, "Dynamic bayesian networks for visual surveillance with distributed cameras," in *Proceedings of the 1st European Conference on Smart Sensing and Context*, pp. 240–243, Enschede, Netherlands, October 2006.

[4] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 34, no. 3, pp. 334–352, 2004.

[5] M. Mallick and B. F. La Scala, "IMM estimator for ground target tracking with variable measurement sampling intervals," in *Proceedings of the 9th International Conference on Information Fusion (FUSION '06)*, pp. 1–8, Florence, Italy, July 2006.

[6] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: a survey," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 103–123, 1998.

[7] L. Di, Y. Yu, and F. He, "Sensor management based on cross-entropy in interacting multiple model kalman filter," in *Proceedings of the 2004 American Control Conference (AAC)*, pp. 5381–5386, Boston, Mass, USA, July 2004.

[8] Z. X. Luo and T. C. Jannett, "Modeling sensor position uncertainty for robust target localization in wireless sensor networks," in *Proceedings of the IEEE Radio and Wireless Symposium*, Santa Clara, Calif, USA, Janurary 2012.

[9] Z. X. Luo and T. C. Jannett, "Energy-based target localization in multi-hop wireless sensor networks," in *Proceedings of the IEEE Radio and Wireless Symposium*, Santa Clara, Calif, USA, Janurary 2012.

[10] Z. X. Luo and T. C. Jannett, "Multi-objective method to balance energy consumption and performance for energy-based target localization in wireless sensor networks," in *Proceedings of the IEEE Southeastcon*, Orlando, Fla, USA, March 2012.

[11] Q. Hao, D. J. Brady, B. D. Guenther, J. B. Burchett, M. Shankar, and S. Feller, "Human tracking with wireless distributed pyroelectric sensors," *IEEE Sensors Journal*, vol. 6, no. 6, pp. 1683–1695, 2006.

[12] K. Xiong, H. Y. Zhang, and C. W. Chan, "Performance evaluation of UKF-based nonlinear filtering," *Automatica*, vol. 42, no. 2, pp. 261–270, 2006.

[13] Y. K. Toh, W. Xiao, and L. Xie, "A wireless sensor network target tracking system with distributed competition based sensor scheduling," in *Proceedings of the 3rd International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP '07)*, pp. 257–262, Melbourne, Australia, December 2007.

[14] W. D. Xiao, J. K. Wu, L. H. Xie, and L. Dong, "Sensor scheduling for target tracking in networks of active sensors," *Acta Automatica Sinica*, vol. 32, no. 6, pp. 922–928, 2006.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 794805, 11 pages
http://dx.doi.org/10.1155/2013/794805

Research Article

On the Joint Time Synchronization and Source Localization Using TOA Measurements

Ming Sun[1] and Le Yang[2]

[1] *Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211, USA*
[2] *School of Internet of Things (IoT) Engineering, Jiangnan University, Wuxi, Jiangsu 214122, China*

Correspondence should be addressed to Le Yang; le.yang.le@gmail.com

Received 1 October 2012; Revised 1 January 2013; Accepted 7 January 2013

Academic Editor: Long Cheng

This paper considers the problem of estimating the clock bias and the position of an unknown source using time of arrival (TOA) measurements obtained at a sensor array to achieve time synchronization and source localization. The study starts with deriving the localization mean square error (MSE) for the case where we pretend that the source clock bias is absent and apply TOA positioning to find the source position. An upper bound on the clock bias, over which we shall obtain a higher localization MSE than that from jointly identifying the clock bias with the source position, is established. Motivated by the MSE analysis, this paper proceeds to develop a new efficient solution for joint synchronization and source localization. The new method is in closed-form, computationally attractive, and more importantly; it is shown analytically to attain the CRLB accuracy under small Gaussian TOA measurement noise. Computer simulations are conducted to corroborate the theoretical development and illustrate the good performance of the proposed algorithm.

1. Introduction

Recently, source localization using wireless sensor networks (WSNs) has attracted great research efforts [1–4]. To determine the source position, positioning parameters such as time of arrival (TOA), time difference of arrival (TDOA), received signal strength (RSS), and angle of arrival (AOA) are commonly used. In this paper, we shall consider determining the source position using TOAs of the source signal received at a wireless sensor network, which is essential for emerging applications including logistics, search and rescue, medical service, smart homes, environmental monitoring, and surveillance [5]. For this problem, a large number of algorithms such as those in [6–14] have been developed in literatures. Nevertheless, almost all of them assumed that the source and the sensors are synchronized ahead of the localization process. In other words, the clock biases of the source and the sensors are known *a priori* so that the obtained TOA measurements can be transformed into range measurement for source localization via TOA positioning.

In practice, the source clock bias may not be readily available and the TOA measurements are therefore subject to unknown source clock bias. In this case, we may simply ignore the source clock bias and still locate the source through TOA positioning. It can be expected by intuition that the obtained source localization accuracy would be worse than the case where the source clock bias is known *a priori*. However, the amount of performance degradation would be small if the source clock bias is negligible compared to the source-sensor distances. On the other hand, another way to handle the presence of the unknown source clock bias is to estimate it jointly with the source position. This would also yield a source localization accuracy poorer than that when the source clock bias is known *a priori*, due to the increase in the number of unknowns to be identified. Clearly, a theoretical analysis is needed here to compare the performance of the above two methods in terms of their localization accuracy. The obtained results would reveal the sensitivity of TOA positioning to the presence of the unknown source clock bias in the TOA measurement. More importantly, they would establish the conditions under which

joint time synchronization and source localization should be preferred in order to achieve better source localization accuracy. To the best of our knowledge, this has not been addressed in existing literatures.

When the source clock bias cannot be ignored and/or it is also of interest besides the source position, an algorithm that can estimate both the source position and clock bias from the TOA measurements is definitely needed. For example, this could occur when a sensor is newly added to the WSN and it needs to be located and synchronized. Conventionally, synchronization and source localization are treated separately. In particular, synchronization is usually achieved first via applying one of the available protocol-domain techniques, such as [15–17], and the localization task can then be accomplished by executing various algorithms, such as those developed in [7–9, 12–14]. A more recent trend is to perform joint synchronization and source localization owing to their close relationship [18–22]. For this problem, given the statistical model of the TOA measurement noise, the maximum likelihood estimator (MLE) can be developed. However, since the cost function is nonconvex, MLE often resorts to the iterative search with a good initial guess for finding a globally optimal solution. This approach possesses the local convergence and even divergence problems. Another disadvantage is the high computational complexity due to possibly large number of iterations. Therefore, a closed-form solution is highly desirable. Bancroft's closed-form solution [23] is the first linearization-based algorithm for 1D timing (i.e., clock bias estimation) and 3D positioning. Zhu and Ding extended Bancroft's method in [21] to the case where the number of TOA measurements is more than the dimension of the unknowns (i.e., the source position plus the clock bias). However, this approach is in general not efficient. In other words, it can attain the Cramér-Rao Lower Bound (CRLB), the best accuracy for an unbiased estimator, in certain scenarios only.

The contribution of this paper is twofold. Firstly, we conduct a mean square error (MSE) analysis to derive the source localization MSE when the source clock bias in the TOA measurements is neglected and TOA positioning is performed to locate the source. For this purpose, a first-order analysis on a pseudo-MLE source location estimator is utilized, which indicates that the obtained results are valid for small TOA noise and source clock bias. The obtained source localization MSE would be partially dependent on the value of the source clock bias and as a result, it reflects the sensitivity of the TOA positioning accuracy to the presence of source clock bias. Moreover, an upper bound on the absolute value of the source clock bias is derived, over which joint synchronization and source localization would provide better localization accuracy in terms of smaller localization MSE than TOA positioning after ignoring the source clock bias. Secondly, this paper proposes a new efficient solution in closed-form for joint synchronization and localization. The newly developed method consists of two processing steps, where the first step locates the source and the second step estimates the clock bias. Compared with the iterative MLE method, the proposed solution is computationally more attractive, since it does not possess local convergence or divergence problem. Moreover, in contrast to the closed-form Bancroft's method [23] and its generalized version [21], the new approach is shown analytically to be able to attain the CRLB accuracy for both source location and source clock under small TOA noise. We support our theoretical developments using extensive computer simulations.

It is worthwhile to point out that the theoretical approach adopted in this paper to perform the MSE analysis and develop the proposed algorithm can be applied in a straightforward manner to the case when the TOAs are deduced from twoway/round-trip time of flight (TOF) measurements. TOF is usually considered as an inexpensive but effective alternative to obtain TOAs without using the cumbersome and costly source-sensor synchronization [24–26]. In this case, however, the turn-around time at the source needs to be either estimated [26] or small enough to be negligible [27, 28] so that TOA positioning can be utilized. With the above observation in mind, it is not difficult to show that the turn-around time in the TOF-deduced TOA measurements can be considered mathematically as an *equivalent* source clock bias. As a result, the MSE analysis and the proposed algorithm presented in this paper can be applied without much modifications to investigate the effect of ignoring the turn-around time and perform joint estimation of the turn-around time and the source position.

The structure of this paper is as follows. Section 2 formulates the joint time synchronization and source localization problem under Gaussian TOA noise model. Besides, the corresponding CRLB is established and analyzed in detail to motivate the MSE analysis and the new estimation algorithm development. Section 3 derives and verifies by computer simulations the localization MSE when the source clock bias in the received TOAs is ignored and the source position is found via TOA positioning. Section 4 presents the proposed closed-form solution for joint synchronization and source localization. Theoretical performance analysis that establishes the efficiency of the new method under small TOA measurement error is also conducted. Section 5 gives the simulation results to illustrate the good performance of the newly developed algorithm, and Section 6 concludes the paper and discusses the future work.

2. Problem Formulation and CRLB

In this section, we shall first formulate the joint time synchronization and source localization problem in consideration. The CRLB for the unknowns is then derived and analysed to motivate the MSE analysis and the algorithm development presented in the following two sections.

2.1. Signal Model. We shall consider jointly synchronizing and locating a single source in a 2D plane. The extensions of the theoretical developments in this paper to the more general case of synchronizing and localizing a source in the 3D space are straightforward.

The source is located at an unknown position $\mathbf{p} = [p_x, p_y]^T$. It has an unknown clock bias of $\tilde{\tau}$ seconds with respect to the reference time (i.e., the *true* time). Mathematically, the local time at the source would be $t - \tilde{\tau}$ if the reference

time is t. The value of $\widetilde{\tau}$ can be negative or positive. Besides, $\widetilde{\tau}$ is unknown but deterministic.

A sensor array composed of M sensors at accurately known positions $\mathbf{p}_m = [p_{m,x}, p_{m,y}]^T$, $m = 1, 2, \ldots, M$, is used to identify $\mathbf{p}$ and $\widetilde{\tau}$ by measuring the TOAs of the source signal. Without loss of generality, we assume that the source starts emitting signals at its local time 0, which corresponds to the reference time $\widetilde{\tau}$. The source signal reaches sensor m at the reference time $\widetilde{\tau} + d_m/c$, where $d_m = \|\mathbf{p} - \mathbf{p}_m\|$ is the true distance between the source and sensor m, $\|\star\|$ denotes the 2-norm and c is the signal propagation speed. Let $\widetilde{\tau}_m$ be the known clock bias of sensor m with respect to the reference time. The TOA measurement obtained at sensor m would be [21], after taking into account the measurement error $\widetilde{n}_m$,

$$\widetilde{T}_m = \widetilde{\tau} - \widetilde{\tau}_m + \frac{d_m}{c} + \widetilde{n}_m. \tag{1}$$

Multiplying both sides of the above equation with the signal propagation speed c yields the TOA equation

$$T_m = \tau - \tau_m + d_m + n_m, \tag{2}$$

where $T_m = c\widetilde{T}_m$, $\tau = c\widetilde{\tau}$, $\tau_m = c\widetilde{\tau}_m$, and $n_m = c\widetilde{n}_m$. Note that the scaled clock biases in (2), namely τ and τ_m, now have the units of meters. Noting that estimating the original source clock bias $\widetilde{\tau}$ is equivalent to identifying its scaled version τ, we shall focus on determining τ in the rest of this paper to simplify the presentation.

Following the noise model in [21], we assume that the TOA measurement errors n_m in (2) are independently and identically distributed (i.i.d.) Gaussian random variables with zero mean and variance σ_n^2. This model has been commonly adopted in literatures such as [10–14] for developing TOA-based localization algorithms and/or studying their performance via computer simulations. Nevertheless, it is a simplification of the real scenario where besides zero-mean Gaussian random errors due to the additive Gaussian noise at every sensor [10], the TOA measurements are also subject to errors owing to the multipath and nonline-of-sight (NLOS) propagation of the source signal as shown in recent experimental studies [29–31]. The fundamental limits of localization accuracy under these realistic factors was established in [32]. Extending our work to more realistic TOA noise models is beyond the scope of this paper, but it is under investigation.

By introducing the $M \times 1$ vectors,

$$\mathbf{T} = \begin{bmatrix} \vdots \\ T_m + \tau_m \\ \vdots \end{bmatrix}, \quad \mathbf{d} = \begin{bmatrix} \vdots \\ d_m \\ \vdots \end{bmatrix}, \quad \mathbf{n} = \begin{bmatrix} \vdots \\ n_m \\ \vdots \end{bmatrix}, \quad \mathbf{1}_M = \begin{bmatrix} \vdots \\ 1 \\ \vdots \end{bmatrix} \tag{3}$$

we can express the measured TOA in (2) using the following signal model in matrix form:

$$\mathbf{T} = \tau \mathbf{1}_M + \mathbf{d} + \mathbf{n}. \tag{4}$$

The problem of interest is to estimate the source position vector $\mathbf{p}$ and the source clock bias τ given $\mathbf{T}$ and $\mathbf{p}_m$.

2.2. CRLB Analysis. The CRLB for the composite unknown vector $\boldsymbol{\theta} = [\tau, \mathbf{p}^T]^T$ under the joint time synchronization and localization scenario presented in the previous subsection is [21]

$$\text{CRLB}(\boldsymbol{\theta}) = \sigma_n^2 \begin{bmatrix} M & \sum_{m=1}^{M} \boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m}^T \\ \sum_{m=1}^{M} \boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m} & \sum_{m=1}^{M} \boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m} \boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m}^T \end{bmatrix}^{-1}, \tag{5}$$

where $\boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m} = (\mathbf{p} - \mathbf{p}_m)/\|\mathbf{p} - \mathbf{p}_m\|$ is a unit vector from $\mathbf{p}_m$ to $\mathbf{p}$. The CRLB of the source clock bias τ, denoted by CRLB(τ), is given by the upper left component of CRLB($\boldsymbol{\theta}$) while the lower right 2×2 block of CRLB($\boldsymbol{\theta}$) is the CRLB of the source position $\mathbf{p}$, denoted by CRLB($\mathbf{p}$). Applying the partitioned matrix inversion formula [33], we have

$$\text{CRLB}(\tau) = \sigma_n^2 \left(M - \sum_{m=1}^{M} \boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m}^T \left(\sum_{m=1}^{M} \boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m} \boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m}^T \right)^{-1} \sum_{m=1}^{M} \boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m} \right)^{-1}, \tag{6a}$$

$$\text{CRLB}(\mathbf{p}) = \sigma_n^2 \left(\sum_{m=1}^{M} \boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m} \boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m}^T - \frac{1}{M} \sum_{m=1}^{M} \boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m} \sum_{m=1}^{M} \boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m}^T \right)^{-1}. \tag{6b}$$

We proceed to evaluate (6b) more carefully to gain more insights. Collecting $\boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m}$, $m = 1, 2, \ldots, M$, yields the $M \times 2$ partial derivative matrix $\mathbf{H}$ given as

$$\mathbf{H} = \begin{bmatrix} \vdots \\ \boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m}^T \\ \vdots \end{bmatrix}. \tag{7}$$

CRLB($\mathbf{p}$) can then be expressed as

$$\begin{aligned} \text{CRLB}(\mathbf{p}) &= \sigma_n^2 \left(\mathbf{H}^T \mathbf{H} - \frac{1}{M} \mathbf{H}^T \mathbf{1}_M \mathbf{1}_M^T \mathbf{H} \right)^{-1} \\ &= \sigma_n^2 \left(\mathbf{H}^T \mathbf{R} \mathbf{H} \right)^{-1}, \end{aligned} \tag{8}$$

where $\mathbf{R} = \mathbf{I}_M - (1/M)\mathbf{1}_M \mathbf{1}_M^T$, $\mathbf{I}_M$ is an $M \times M$ identity matrix and $\mathbf{1}_M$ is an $M \times 1$ column vector whose elements are all equal to one as defined in (4). The matrix $\mathbf{R}$ can be rewritten as the following matrix product:

$$\begin{aligned} \mathbf{R} = &\begin{bmatrix} -\mathbf{1}_{M-1}^T \\ \mathbf{I}_{M-1} \end{bmatrix} \left(\mathbf{I}_{M-1} - \frac{1}{M} \mathbf{1}_{M-1} \mathbf{1}_{M-1}^T \right) \\ &\times \begin{bmatrix} -\mathbf{1}_{M-1} & \mathbf{I}_{M-1} \end{bmatrix}, \end{aligned} \tag{9}$$

where $\mathbf{I}_{M-1}$ is an $(M-1)\times(M-1)$ identity matrix and $\mathbf{1}_{M-1}$ is an $(M-1)\times 1$ column vector whose elements are all equal to one. Applying the matrix inversion lemma [33] to the matrix in the middle of the right hand sides of (9) yields

$$\left(\mathbf{I}_{M-1}-\frac{1}{M}\mathbf{1}_{M-1}\mathbf{1}_{M-1}^{T}\right)=\left(\mathbf{I}_{M-1}+\mathbf{1}_{M-1}\mathbf{1}_{M-1}^{T}\right)^{-1}=\mathbf{Q}_{\alpha}^{-1}. \tag{10}$$

Clearly, $\mathbf{Q}_{\alpha}$ is an $(M-1)\times(M-1)$ matrix with all its diagonal elements equal to 2 and others equal to 1. Substituting (10) back into (9) and putting the result into (8), we have

$$\mathrm{CRLB}\,(\mathbf{p})=\sigma_n^2\left(\widetilde{\mathbf{H}}^{T}\mathbf{Q}_{\alpha}^{-1}\widetilde{\mathbf{H}}\right)^{-1}, \tag{11}$$

where $\widetilde{\mathbf{H}}=[-\mathbf{1}_{M-1}\ \ \mathbf{I}_{M-1}]\mathbf{H}$ is an $(M-1)\times 2$ matrix whose $(m-1)$th row, $m=2,3,\ldots,M$, is equal to

$$\widetilde{\mathbf{H}}\,(m-1,:)=\boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m}^{T}-\boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_1}^{T}. \tag{12}$$

Comparing (11) with (33) from [34], we notice that within the considered time synchronization and source localization framework, the CRLB of the source position is equal to that of locating the source from $M-1$ time difference of arrival (TDOA) measurements $r_{m1}=(T_m+\tau_m)-(T_1+\tau_1)$, where $m=2,3,\ldots,M$. In other words, the TDOA measurements r_{m1} are indeed generated by subtracting the TOA obtained at sensor 1 from the remaining TOAs. From the definition of the TOA measurements T_m given in (2), all the TOAs are linearly related to the unknown source clock bias τ and hence, the subtraction operation eliminates the presence of τ in the obtained TDOAs r_{m1}. This enables TDOA positioning of the source and also forms the basis of the joint synchronization and localization algorithm proposed in Section 4 of this paper.

It is worthwhile to point out that we choose to produce the TDOAs through subtracting the TOA measured at sensor 1 from the rest $M-1$ TOAs for two reasons. First, this facilitates the comparison of the CRLB result derived in (11) with (33) from [34], where the TDOA measurements were also obtained with respect to sensor 1. Second, subtracting the TOA at any sensor other than sensor 1 from the remaining $M-1$ TOAs would yield a different set of TDOA measurements but with the same quality as r_{m1} in terms of the source localization CRLB. This can be verified by applying the fact from Section 2 that the TOA measurement noises n_m are i.i.d. and they have the same variance. Nevertheless, the above observation would become invalid if n_m are no longer i.i.d. Investigating the approach that takes into account the statistical information on n_m to produce the optimal TDOA measurement set is an important topic subject to future researches.

When the clock bias of the source, τ, is available *a priori,* but the source position $\mathbf{p}$ remains unknown, the joint synchronization and localization task reduces to the classic problem of TOA positioning using the measurements in (2). The best possible localization accuracy for this case becomes [21]

$$\mathrm{CRLB}(\mathbf{p})_o=\sigma_n^2\left(\sum_{m=1}^{M}\boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m}\boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m}^{T}\right)^{-1}. \tag{13}$$

$\mathrm{CRLB}(\mathbf{p})_o$ and $\mathrm{CRLB}(\mathbf{p})$ given in (6b) are the CRLBs of the source location estimation under two different scenarios, namely, when the source block bias τ is known and when τ is unknown. They are both 2×2 matrices and have the same units, which makes subtracting $\mathrm{CRLB}(\mathbf{p})_o$ in (13) from $\mathrm{CRLB}(\mathbf{p})$ in (6b) feasible. In fact, it can shown that $\mathrm{CRLB}(\mathbf{p})$-$\mathrm{CRLB}(\mathbf{p})_o$ is a positive semidefinite matrix, which indicates that compared with TOA positioning, the problem of joint time synchronization and source localization considered in this paper has a worse localization accuracy in general. This is expected, because we need to estimate one more unknown, specifically the source clock bias τ, from the same set of TOA measurements.

On the other hand, carefully examining (2) reveals that if the clock bias τ is negligible compared to the source-sensor distance d_m, $m=1,2,\ldots,M$, or equivalently $\tau/d_m\approx 0$, and the source position is of primary interest, we may simply ignore the presence of τ and apply TOA positioning technique to the TOA measurements T_m in (2) for identifying $\mathbf{p}$. In this way, the obtained location estimate will be biased, but it could have a smaller localization MSE than that of the location estimate from an efficient estimator for joint time synchronization and source localization. We shall theoretically illustrate the above observations in the following section.

3. Localization MSE Analysis

In this section, we shall derive the localization MSE when identifying the source position $\mathbf{p}$ from the TOA measurements T_m in (2) via TOA positioning and pretending that the source clock bias τ is zero. The obtained results will be contrasted with the best achievable localization MSE under the framework of joint estimation of τ and $\mathbf{p}$, which is equal to the trace of the $\mathrm{CRLB}(\mathbf{p})$ given in (6b). The theoretical developments are based on an estimator that utilizes the Taylor-series linearization and estimates the source location from T_m via the classic weighted least squares (WLS) technique. The reason behind the use of a WLS source location estimate is that it enables obtaining its localization MSE in closed form in terms of the TOA measurement noise power σ_n^2 and the source clock bias τ to gain more insights. The localization MSE result in this section is valid for the scenario where the TOA measurement noise and the source clock bias are both small, due to the use of first-order approximation in the considered location estimator. But it applies to any TOA positioning technique that achieves the CRLB accuracy (13) when the source clock bias is known *a priori*. We shall corroborate the analytical results using numerical examples in Section 3.1 where the maximum-likelihood (ML) estimator for TOA positioning that attains (13) asymptotically when the source clock bias is known is simulated.

The MSE analysis starts with approximating the TOA measurement in (2) via applying the Taylor-series expansion up to the linear term to the source-sensor distance $d_m=\|\mathbf{p}-\mathbf{p}_m\|$ using an initial source position guess $\widetilde{\mathbf{p}}$. Mathematically, we have

$$T_m\approx\tau-\tau_m+\left\|\widetilde{\mathbf{p}}-\mathbf{p}_m\right\|+\boldsymbol{\rho}_{\widetilde{\mathbf{p}},\mathbf{p}_m}^{T}\left(\mathbf{p}-\widetilde{\mathbf{p}}\right). \tag{14}$$

It can be deduced from (14) that the identification of the source position $\mathbf{p}$ is equivalent to find $(\mathbf{p}-\tilde{\mathbf{p}})$. For this purpose, we rearrange (14) and stack the result for $m = 1, 2, \ldots, M$ to arrive at

$$\underbrace{\begin{bmatrix} \vdots \\ T_m + \tau_m - \|\tilde{\mathbf{p}} - \mathbf{p}\| \\ \vdots \end{bmatrix}}_{\mathbf{h}} = \underbrace{\begin{bmatrix} \vdots \\ \boldsymbol{\rho}_{\tilde{\mathbf{p}},\mathbf{p}_m}^T \\ \vdots \end{bmatrix}}_{\mathbf{G}} (\mathbf{p} - \tilde{\mathbf{p}}) + \tau \mathbf{1}_M + \mathbf{n}, \quad (15)$$

where $\mathbf{1}_M$ and $\mathbf{n}$ are defined in (4). Pretending that $\tau = 0$ and utilizing $\mathbf{n}$ being zero-mean Gaussian random vector, the maximum likelihood (ML) estimate for $(\mathbf{p}-\tilde{\mathbf{p}})$ can be derived using the WLS technique [35]. Adding the result to the initial solution guess for $\mathbf{p}$ yields the final source position estimate

$$\tilde{\tilde{\mathbf{p}}} = \tilde{\mathbf{p}} + \left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\mathbf{h}. \quad (16)$$

Substituting (15) into the above equation and subtracting the true source position $\mathbf{p}$ from both sides give the estimation error, when the clock bias τ is ignored

$$\tilde{\tilde{\mathbf{p}}} - \mathbf{p} = \left(\mathbf{G}^T\mathbf{G}\right)^{-1}\mathbf{G}^T\left(\tau\mathbf{1}_M + \mathbf{n}\right). \quad (17)$$

Assuming that the initial source position guess $\tilde{\mathbf{p}}$ is sufficiently close to the true value $\mathbf{p}$, we can ignore the error in $\mathbf{G}$ and approximate it with $\mathbf{H}$ defined in (7) whose row vectors are $\boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m}^T$. Therefore, (17) becomes

$$\tilde{\tilde{\mathbf{p}}} - \mathbf{p} \approx \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\left(\tau\mathbf{1}_M + \mathbf{n}\right). \quad (18)$$

It can been observed that $\tilde{\tilde{\mathbf{p}}}$ is a biased estimate of $\mathbf{p}$ and the estimation bias is equal to $\tau(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{1}_M$, which is proportional to the value of the source clock bias τ. Moreover, the localization MSE can be derived by multiplying both sides of (18) with the transpose of $\tilde{\tilde{\mathbf{p}}} - \mathbf{p}$ and taking expectation as well as the matrix trace. We have

$$\begin{aligned} \mathrm{MSE}\left(\tilde{\tilde{\mathbf{p}}}\right) = {} & \mathrm{tr}\Big(\sigma_n^2\left(\mathbf{H}^T\mathbf{H}\right)^{-1} \\ & + \tau^2\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{1}_M\mathbf{1}_M^T\mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\Big), \end{aligned} \quad (19)$$

where tr() is the trace of a matrix.

On the other hand, the localization MSE of an efficient estimator for jointly identifying $\mathbf{p}$ and τ can be derived by applying the matrix inversion lemma [33] to (8) and evaluating its trace. Subtracting the result from (19) yields

$$\begin{aligned} & \mathrm{MSE}\left(\tilde{\tilde{\mathbf{p}}}\right) - \mathrm{tr}\left(\mathrm{CRLB}\left(\mathbf{p}\right)\right) \\ & = \left(\tau^2 - \frac{\sigma_n^2}{M - \mathbf{1}_M^T\mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{1}_M}\right) \\ & \quad \times \mathrm{tr}\left(\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{1}_M\mathbf{1}_M^T\mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\right). \end{aligned} \quad (20)$$

This indicates that ignoring the clock bias τ may still lead to a location estimate with a smaller localization MSE than that of jointly identifying it together with the source position $\mathbf{p}$, if the following inequality holds:

$$\tau^2 < \frac{\sigma_n^2}{M - \mathbf{1}_M^T\mathbf{H}\left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{1}_M}. \quad (21)$$

The term on the right hand side of (21) is dependent on the TOA measurement noise power and the localization geometry. In particular, it increases with σ_n^2. Furthermore, it would also become relatively large when the source lies away from the sensor array. This is because in this case, the matrix inverse $(\mathbf{H}^T\mathbf{H})^{-1}$ in the denominator, which is in fact the scaled version of the best TOA localization accuracy $\mathrm{CRLB}(\mathbf{p})_o$ when the clock bias is known (see (13)), would increase. On the other hand, the term on the right hand side of (21) would decrease if $\mathbf{H}^T\mathbf{1}_M$ is close to a zero vector. This could occur when the source is inside the sensor array.

3.1. Numerical Examples. We shall verify the theoretical analysis results via computer simulations. The considered scenario is depicted in Figure 1, where $M = 6$ sensors are uniformly deployed along a circle centered at the origin and having a radius of 20m. The source is located either at $\mathbf{p} = [50, 50]^T$m which is outside the sensor array or at $\mathbf{p} = [-10, 10]^T$m which is inside the sensor array.

The MLE is realized to identify the source position and its localization MSE is defined as $(1/L)\sum_{l=1}^{L}\|\mathbf{p}_l - \mathbf{p}\|^2$, where $\mathbf{p}_l$ is the source location estimate in the lth ensemble run and $L = 20,000$ is the total number of ensemble runs. The noisy TOA measurements fed to the MLE in each ensemble run are generated by adding to the true values zero-mean Gaussian noise n_m with variance σ_n^2 set to be -20 in log scale. The source position is found via maximizing the *pseudo*-ML cost function $(\mathbf{T} - \mathbf{d})^T(\mathbf{T} - \mathbf{d})$ with respect to the source position $\mathbf{p}$, where $\mathbf{T}$ and $\mathbf{d}$ are both defined in (4), $\mathbf{T}$ collects the TOA measurements (having clock bias) and $\mathbf{d}$ is functionally dependent on $\mathbf{p}$. In this way, the MLE performs the TOA-localization task by pretending that the clock bias τ in $\mathbf{T}$ is absent. The maximization is conducted via applying the iterative Taylor-series method [36] with the initial solution guess produced by adding to the true source position zero-mean Gaussian random vector with a covariance matrix of $2\mathrm{CRLB}(\mathbf{p})$, where $\mathrm{CRLB}(\mathbf{p})$ is defined in (6b). This ensures that the MLE would normally need several iterations to converge to a solution.

Two sets of simulation results are generated, one for the source outside the sensor array and the other for the source inside the sensor array. They are plotted as function of the source clock bias τ in Figures 2 and 3. Also included in the figures are the theoretical localization MSE when ignoring the clock bias, which is given in (19). We plot as well the traces of $\mathrm{CRLB}(\mathbf{p})$ in (6b) and $\mathrm{CRLB}(\mathbf{p})_o$ in (13). They denote the lowest possible localization MSEs when the clock bias τ is jointly estimated with the source position and when τ is known *a priori*.

We can see from Figure 2 that when the source clock bias τ is zero, the localization MSE of the simulated ML estimator reaches $\mathrm{CRLB}(\mathbf{p})_o$ as expected. As τ increases, the localization

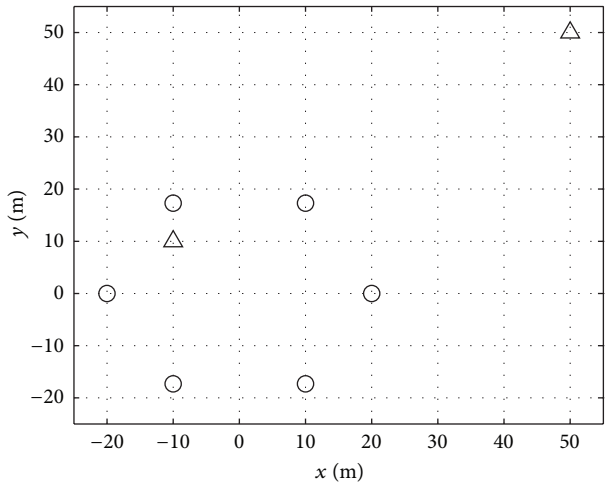


Figure 1: Joint synchronization and source localization scenario. The sensors are denoted by circle symbols while the upper triangle symbols represent the two true source positions considered.

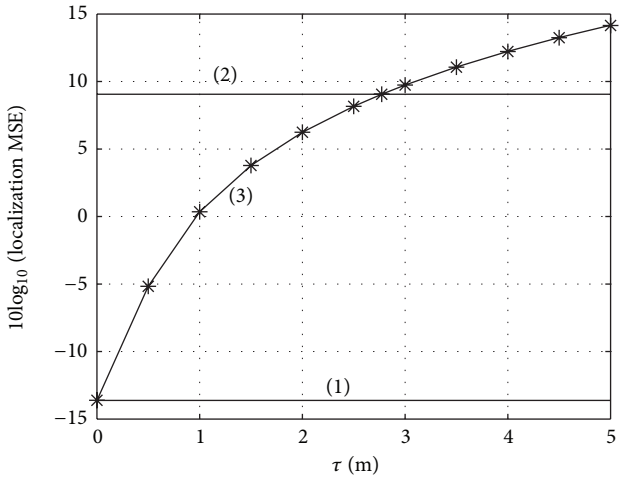


Figure 2: Localization accuracy for the source outside the sensor array when ignoring the presence of clock bias. (1) tr(CRLB($\mathbf{p}$)$_o$) from (13), (2) tr(CRLB($\mathbf{p}$)) from (6b), (3) MSE($\widetilde{\mathbf{p}}$) from (19), star symbol: source localization MSE from the simulated pseudo-ML estimator.

performance of the considered MLE degrades as predicted in (19). The simulation MSE matches the theoretical results very well, which justifies the validity of the localization MSE analysis. Furthermore, the localization MSE of the MLE remains to be smaller than that of jointly identifying the source clock bias and the source position, until the clock bias reaches a certain level (in this simulation, around $\tau = 2.77$ m). This is consistent with the analysis in (21). In Figure 2, the boundary condition (i.e., the crossing point of curves (2) and (3)) that makes (21) achieve equality is also verified through simulation. Curves (2) and (3) are indeed the localization MSEs when the source is located jointly with the source clock bias and when the source position is found via ignoring the source clock bias and applying TOA positioning. The existence of a crossing point indicates that the two techniques can yield the same performance in terms of localization MSE for a particular value of the source clock bias.

The observations from Figure 3 where the source is inside the sensor array are very similar to those obtained from Figure 2 where the source is outside the sensor array. An important difference is that in this case, the crossing point of curves (2) and (3) appears when the source clock bias is equal to $\tau = 0.0492$ m. This observation is consistent with the analysis under (21). The crossing point being much closer to the origin compared with Figure 2 indicates that when the source lies inside the sensor array, ignoring the source clock bias and locating the source via TOA positioning is more likely to produce worse localization performance than joint clock bias and source position estimation, considering the presence of clock asynchronization (see, e.g., [32]). It is worthwhile to point out that the above observation is obtained without taking into account other realistic factors such as the multipath effect in signal propagation. We are currently extending our theoretical developments to more realistic signal models.

It can also be seen from Figure 3 that when the clock bias τ is as low as 0.05 m, which corresponds to around

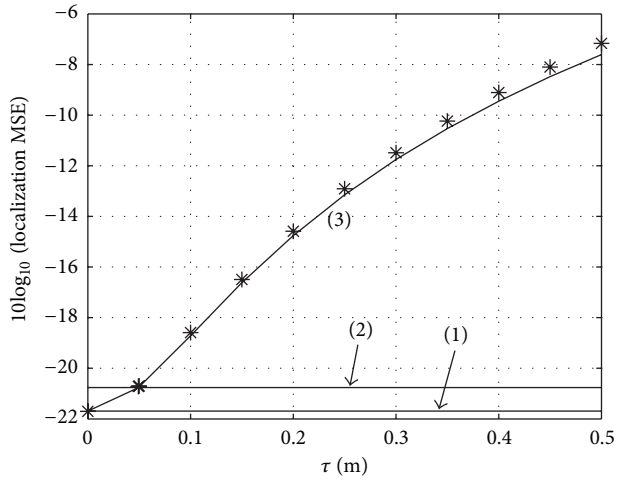


Figure 3: Localization accuracy for the source inside the sensor array when ignoring the presence of clock bias. (1) tr(CRLB($\mathbf{p}$)$_o$) from (13), (2) tr(CRLB($\mathbf{p}$)) from (6b), (3) MSE($\widetilde{\mathbf{p}}$) from (19), star symbol: source localization MSE from the simulated pseudo-ML estimator.

0.17 nanoseconds, if the signal propagates at the speed of light, ignoring τ leads to an amount of more than 10 dB degradation in source localization MSE, compared to the joint time synchronization and localization. With those observations in mind and noting that existing closed-form joint synchronization and localization techniques, such as the one proposed in [21], are not able to provide the CRLB accuracy, we are motivated to develop in the next section a novel algorithm in closed-form that can identify τ and the source position $\mathbf{p}$ efficiently.

4. Algorithm and Performance Analysis

In this section, we shall develop an efficient algorithm for jointly estimating the clock bias τ and the source position $\mathbf{p}$ from the M TOA measurements in (4). The algorithm is in closed-form and has low computational complexity. It consists of two processing steps, where Step 1 locates the source and Step 2 estimates τ. We shall show analytically that under small TOA measurement noise, the proposed algorithm reaches the CRLB accuracy for both the source position and the clock bias.

Step 1. The development of Step 1 processing is motivated by the CRLB analysis presented in Section 2.2. In particular, we have shown there that the CRLB of the source position, when jointly identified with the clock bias, is identical to that under TDOA positioning, where the $M-1$ TDOAs are produced by subtracting the TOA measurement obtained at sensor 1 from the remaining $M-1$ TOA measurements. Following this observation, we locate the source position $\mathbf{p}$ as follows. First, we obtain the TDOA measurements via

$$r_{m1} = \left(T_m + \tau_m\right) - \left(T_1 + \tau_1\right), \quad m = 2, 3, \ldots, M. \tag{22}$$

From (2), the measurement errors in r_{m1} are $n_{m1} = n_m - n_1$. Collecting r_{m1} forms the $(M-1)\times 1$ column vector $\mathbf{r} = [r_{21}, r_{31}, \ldots, r_{M1}]^T$ and its error vector is correspondingly

$$\Delta\mathbf{r} = \left[n_{21}, n_{31}, \ldots, n_{M1}\right]^T = \left[-\mathbf{1}_{M-1} \quad \mathbf{I}_{M-1}\right]\mathbf{n}, \tag{23}$$

where $\mathbf{n}$ is the TOA measurement noise vector defined in (4). $\Delta\mathbf{r}$ is a zero-mean random vector because $\mathbf{n}$ have zero mean and its covariance matrix is $\sigma_n^2\mathbf{Q}_\alpha$, where $\mathbf{Q}_\alpha$ is defined in (10). Next, the source position $\mathbf{p}$ is estimated from $\mathbf{r}$ using the closed-form twostage algorithm developed in [34]. We shall briefly summarize the computations. Interested readers can find the algorithm details in [34].

Stage 1 of Step 1 processing estimates $\mathbf{p}$ together with an extra variable $d_1 = \|\mathbf{p} - \mathbf{p}_1\|$ that is the distance between the source and sensor 1. The functional relationship between them is ignored in Stage 1. Define the unknown vector as $\boldsymbol{\varphi}_1 = [\mathbf{p}^T, d_1]^T$ and its weighted least squares (WLS) estimate is found via

$$\widehat{\boldsymbol{\varphi}}_1 = \left(\mathbf{G}_1^T\mathbf{W}_1\mathbf{G}_1\right)^{-1}\mathbf{G}_1^T\mathbf{W}_1\mathbf{h}_1, \tag{24}$$

where the weighting matrix $\mathbf{W}_1$ is equal to $\mathbf{W}_1 = (\mathbf{B}_1\mathbf{Q}_\mathbf{a}\mathbf{B}_1^T)^{-1}/\sigma_n^2$, $\mathbf{B}_1 = \mathrm{diag}([d_2, d_3, \ldots, d_M])$ and d_m, $m = 2, 3, \ldots, M$, are the ranges between the source and sensor m (see (2)). The matrix $\mathbf{G}_1$ and the column vector $\mathbf{h}_1$ both have $(M-1)$ rows and their $(m-1)$th rows are $\mathbf{G}_1(m-1,:) = -2\cdot[(\mathbf{p}_m - \mathbf{p}_1)^T, r_{m1}]$ and $\mathbf{h}_1(m-1) = r_{m1}^2 - \mathbf{p}_m^T\mathbf{p}_m + \mathbf{p}_1^T\mathbf{p}_1$. The estimation error is equal to $\Delta\widehat{\boldsymbol{\varphi}}_1 = (\mathbf{G}_1^T\mathbf{W}_1\mathbf{G}_1)^{-1}\mathbf{G}_1^T\mathbf{W}_1\mathbf{B}_1\Delta\mathbf{r}$ which is approximately zero-mean (under small noise condition and $\mathbf{G}_1$ being approximately noise-free) and has a covariance matrix $\mathrm{cov}(\widehat{\boldsymbol{\varphi}}_1) = (\mathbf{G}_1^T\mathbf{W}_1\mathbf{G}_1)^{-1}$.

Stage 2 of Step 1 processing refines the source position estimate from Stage 1, which is $\widehat{\boldsymbol{\varphi}}_1(1:2)$, by exploring the relation between the unknowns $\mathbf{p}$ and d_1. The unknown vector of this stage is $\boldsymbol{\varphi}_2 = (\mathbf{p} - \mathbf{p}_1)\odot(\mathbf{p} - \mathbf{p}_1)$, where $\odot$ is the Schur product (element by element multiplication). Its estimate is again found via applying the WLS technique, which is given by

$$\widehat{\boldsymbol{\varphi}}_2 = \left(\mathbf{G}_2^T\mathbf{W}_2\mathbf{G}_2\right)^{-1}\mathbf{G}_2^T\mathbf{W}_2\mathbf{h}_2, \tag{25}$$

where the weighting matrix is $\mathbf{W}_2 = (\mathbf{B}_2\,\mathrm{cov}(\widehat{\boldsymbol{\varphi}}_1)\mathbf{B}_2^T)^{-1}$ and $\mathbf{B}_2 = 2\cdot\mathrm{diag}([(\mathbf{p} - \mathbf{p}_1)^T, d_1])$. $\mathbf{G}_2$ and $\mathbf{h}_2$ are equal to $\mathbf{G}_2 = [\mathbf{I}_2, \mathbf{1}_2]^T$ and $\mathbf{h}_2 = [(\widehat{\boldsymbol{\varphi}}_1(1:2) - \mathbf{p}_1)^T \odot (\widehat{\boldsymbol{\varphi}}_1(1:2) - \mathbf{p}_1)^T, \widehat{\boldsymbol{\varphi}}_1^2(3)]^T$, where $\mathbf{I}_2$ is a 2×2 identity matrix and $\mathbf{1}_2$ is a 2×1 column vector of ones. The estimation error in $\widehat{\boldsymbol{\varphi}}_2$ is $\Delta\widehat{\boldsymbol{\varphi}}_2 = (\mathbf{G}_2^T\mathbf{W}_2\mathbf{G}_2)^{-1}\mathbf{G}_2^T\mathbf{W}_2\mathbf{B}_2\Delta\widehat{\boldsymbol{\varphi}}_1$ with a covariance matrix of $\mathrm{cov}(\widehat{\boldsymbol{\varphi}}_2) = (\mathbf{G}_2^T\mathbf{W}_2\mathbf{G}_2)^{-1}$.

The source position estimate is obtained through mapping $\widehat{\boldsymbol{\varphi}}_2$ using

$$\widehat{\mathbf{p}} = \mathrm{diag}\left(\left[\mathrm{sgn}\left(\widehat{\boldsymbol{\varphi}}_1(1:2) - \mathbf{p}_1\right)\right]\right)\sqrt{\widehat{\boldsymbol{\varphi}}_2} + \mathbf{p}_1, \tag{26}$$

where $\mathrm{sgn}(x)$ is a sign function equal to 1 for $x \geq 0$ and 0 otherwise. The source localization error would be $\Delta\widehat{\mathbf{p}} = \mathbf{B}_3^{-T}\Delta\widehat{\boldsymbol{\varphi}}_2$, where $\mathbf{B}_3 = 2\cdot\mathrm{diag}([\mathbf{p}^T - \mathbf{p}_1^T])$. The covariance matrix of $\widehat{\mathbf{p}}$ can be shown to be $\mathrm{cov}(\widehat{\mathbf{p}}) = \mathbf{B}_3^{-T}\,\mathrm{cov}(\widehat{\boldsymbol{\varphi}}_2)\mathbf{B}_3^{-1}$.

Step 2. Step 2 estimates the clock bias τ. In particular, under the joint synchronization and localization problem formulated in Section 2.1 (see (4)), the maximum likelihood (ML) estimates of τ and the source position $\mathbf{p}$ can be found via solving the following minimization problem:

$$\min_{\tau,\mathbf{p}}\left(\mathbf{T} - \mathbf{1}_M\tau - \mathbf{d}\right)^T\left(\mathbf{T} - \mathbf{1}_M\tau - \mathbf{d}\right), \tag{27}$$

where the fact that the TOA measurement noise vector $\mathbf{n}$ is a zero-mean Gaussian random vector has been applied. Differentiating the cost function in (27) with respect to τ and setting the result to zero yield the ML estimator for the clock bias τ. Mathematically, we have

$$\widehat{\tau} = \frac{1}{M}\mathbf{1}_M^T\left(\mathbf{T} - \mathbf{d}\right). \tag{28}$$

$\mathbf{d}$ is defined in (4) and it is unknown because the true source position $\mathbf{p}$ is not available. We replace it with $\widehat{\mathbf{d}}$ that has the same functional form as $\mathbf{d}$ except that $\mathbf{p}$ has been replaced by Step 1 output $\widehat{\mathbf{p}}$. Hence, the proposed algorithm estimates the clock bias via

$$\widehat{\tau} = \frac{1}{M}\mathbf{1}_M^T\left(\mathbf{T} - \widehat{\mathbf{d}}\right). \tag{29}$$

We now summarize the processing required to estimate the source position $\mathbf{p}$ and the clock bias τ. The proposed algorithm accomplishes the joint synchronization and source localization task by evaluating sequentially (24), (25), (26), and (29). Note that in computing $\widehat{\boldsymbol{\varphi}}_1$ and $\widehat{\boldsymbol{\varphi}}_2$ using (24) and (25), the true source position $\mathbf{p}$ is needed to produce the weighting matrices $\mathbf{W}_1$ and $\mathbf{W}_2$. To bypass these difficulties,

when evaluating (24), we first set $\mathbf{W}_1$ to be an identity matrix of appropriate dimension and compute $\widehat{\boldsymbol{\varphi}}_1$ to find $\boldsymbol{\varphi}_1$. Then, (24) is calculated again with $\mathbf{p}$ in $\mathbf{W}_1$ being replaced by $\widehat{\boldsymbol{\varphi}}_1(1:2)$ that is just obtained. In evaluating (25), the source position estimate from Stage 1 of Step 1 processing is utilized to generate $\mathbf{W}_2$.

4.1. Performance Analysis. We shall establish the efficiency of the algorithm proposed in the previous subsection under the condition that the TOA measurement noise n_m are sufficiently small. Mathematically, we need to show that the algorithm output, namely the source position estimate $\widehat{\mathbf{p}}$ and the clock bias estimate $\widehat{\tau}$, is unbiased and their covariance matrices are approximately equal to the corresponding CRLBs, that is,

$$\operatorname{cov}(\widehat{\mathbf{p}}) \approx \operatorname{CRLB}(\mathbf{p}), \tag{30}$$

$$\operatorname{cov}(\widehat{\tau}) \approx \operatorname{CRLB}(\tau). \tag{31}$$

$\operatorname{cov}(\widehat{\mathbf{p}})$ is the covariance matrix of $\widehat{\mathbf{p}}$ given under (26) and $\operatorname{CRLB}(\mathbf{p})$ is the CRLB of the source position $\mathbf{p}$ defined in (6b). The validity of (30) can be verified by following the performance analysis procedure adopted in [34]. The unbiasedness of $\widehat{\mathbf{p}}$ can be shown as follows. The estimation error in $\widehat{\mathbf{p}}$ can be written as, from the definitions of $\Delta\widehat{\mathbf{p}}$, $\Delta\widehat{\boldsymbol{\varphi}}_2$ and $\Delta\widehat{\boldsymbol{\varphi}}_1$ given below (26), (25), and (24),

$$\Delta\widehat{\mathbf{p}} = \underbrace{\mathbf{B}_3^{-T}\left(\mathbf{G}_2^T\mathbf{W}_2\mathbf{G}_2\right)^{-1}\mathbf{G}_2^T\mathbf{W}_2\mathbf{B}_2\left(\mathbf{G}_1^T\mathrm{W}_1\mathbf{G}_1\right)^{-1}\mathbf{G}_1^T\mathbf{W}_1\mathbf{B}_1}_{\mathbf{U}}\Delta\mathbf{r}, \tag{32}$$

where $\Delta\mathbf{r}$ is the error in the TDOA measurement vector produced from the TOA measurements T_m, which is defined in (23) and shown to have zero mean. Under the small TOA measurement noise condition, we can ignore the noise in $\mathbf{G}_1$ and as a result, $\Delta\widehat{\mathbf{p}}$ is linearly proportional to $\Delta\mathbf{r}$ and is zero-mean, which establishes that the source location estimate from the proposed algorithm is unbiased.

We proceed to prove (31), where $\operatorname{cov}(\widehat{\tau})$ is the covariance matrix of the clock bias estimate in (29) and $\operatorname{CRLB}(\tau)$ is the CRLB defined in (6a). As pointed out above (29), the clock bias is identified within the proposed algorithm using an ML estimator. We have shown above that the source position estimate $\widehat{\mathbf{p}}$ is an efficient estimate and as such, it can be expected from the property of the ML estimator [35] that the clock bias estimate $\widehat{\tau}$ would have an accuracy approximately equal to its CRLB. To derive $\operatorname{cov}(\widehat{\tau})$, the estimation error in $\widehat{\tau}$, denoted by $\Delta\widehat{\tau}$, needs to be found. For this purpose, we expand $\widehat{\mathbf{d}}$ in (29) at the true source position $\mathbf{p}$ using the Taylor-series expansion up to the linear term, substitute (4), and subtract τ from both sides of (29) to arrive at

$$\Delta\widehat{\tau} \approx \frac{1}{M}\mathbf{1}_M^T\left(\mathbf{n} - \mathbf{H}\Delta\widehat{\mathbf{p}}\right), \tag{33}$$

where $\mathbf{H}$ is defined in (7). It can be shown by putting (32) into (33) that the clock bias estimate $\widehat{\tau}$ from the proposed algorithm is also zero-mean since its estimation error $\Delta\widehat{\tau}$ is linearly proportional to the zero-mean TOA measurement noise vector $\mathbf{n}$, under small TOA noise condition.

$\operatorname{cov}(\widehat{\tau})$ can be obtained via squaring both sides of (33) and taking expectation. We have, after some simplifications,

$$\operatorname{cov}(\widehat{\tau}) \approx \frac{1}{M^2}\mathbf{1}_M^T\left(\sigma_n^2\mathbf{I}_M + \mathbf{H}\operatorname{CRLB}(\mathbf{p})\mathbf{H}^T - \mathbf{F} - \mathbf{F}^T\right)\mathbf{1}_M, \tag{34}$$

where (30) has been substituted and $\mathbf{F} = \mathbf{H}\cdot\mathrm{E}[\Delta\widehat{\mathbf{p}}\mathbf{n}^T]$. We shall show that $\mathbf{F}\cdot\mathbf{1}_M = \mathbf{0}_M$, where $\mathbf{0}_M$ is an $M\times 1$ vector of zeros. In particular, after putting (32) and (23),

$$\begin{aligned}\mathbf{F}\cdot\mathbf{1}_M &\approx \mathbf{H}\mathbf{U}\cdot\mathrm{E}\left[\Delta\mathbf{r}\mathbf{n}^T\right]\mathbf{1}_M \\ &= \mathbf{H}\mathbf{U}\cdot\left[-\mathbf{1}_{M-1}\quad \mathbf{I}_{M-1}\right]\mathbf{1}_M = \mathbf{0}_M.\end{aligned} \tag{35}$$

Putting the above result into (34) and comparing (34) with (6a) after applying the matrix inversion lemma [33] and $\sum_{m=1}^{M}\boldsymbol{\rho}_{\mathbf{p},\mathbf{p}_m}^T = \mathbf{1}_M^T\mathbf{H}$, which is

$$\begin{aligned}\operatorname{CRLB}(\tau) &= \sigma_n^2\left(\frac{1}{M} + \frac{1}{M^2}\mathbf{1}_M^T\mathbf{H}\right. \\ &\qquad\left.\times\left(\mathbf{H}^T\mathbf{H} - \frac{1}{M}\mathbf{H}^T\mathbf{1}_M\mathbf{1}_M^T\mathbf{H}\right)^{-1}\mathbf{H}^T\mathbf{1}_M\right) \\ &= \left(\frac{\sigma_n^2}{M} + \frac{1}{M^2}\mathbf{1}_M^T\mathbf{H}\operatorname{CRLB}(\mathbf{p})\mathbf{H}^T\mathbf{1}_M\right)\end{aligned} \tag{36}$$

would yield (31). This completes the establishment of the efficiency of the proposed algorithm.

5. Simulations

We shall demonstrate the estimation performance of the algorithm developed in the previous section for identifying the position and the clock bias of the unknown source in a twostep manner via computer simulations. The simulation scenario is the same as in Section 3.1, which is depicted in Figure 1.

The newly proposed algorithm in Section 4 is applied to estimate the source position $\mathbf{p}$ as well as the clock bias fixed at $\tau = 15$ m. The accuracy for source localization and synchronization is quantified using $(1/L)\sum_{l=1}^{L}\|\mathbf{p}_l - \mathbf{p}\|^2$ and $(1/L)\sum_{l=1}^{L}\|\tau_l - \tau\|^2$, where $L = 20{,}000$ is the total number of ensemble runs and $\mathbf{p}_l$ and τ_l are the source position and clock bias estimates in ensemble l. In each ensemble run, the erroneous TOA measurements are produced by adding to the true values zero-mean Gaussian noise with variance σ_n^2. For the purpose of comparison, we also realize the algorithm developed in [21] for joint source localization and synchronization.

For each of the two sources shown in Figure 1, we generate two figures for plotting the synchronization and source localization MSEs as function of the TOA measurement noise power σ_n^2, respectively. The corresponding CRLBs, specifically $\operatorname{CRLB}(\tau)$ and $\operatorname{CRLB}(\mathbf{p})$ from (6a) and (6b), are included in the figures as performance benchmark.

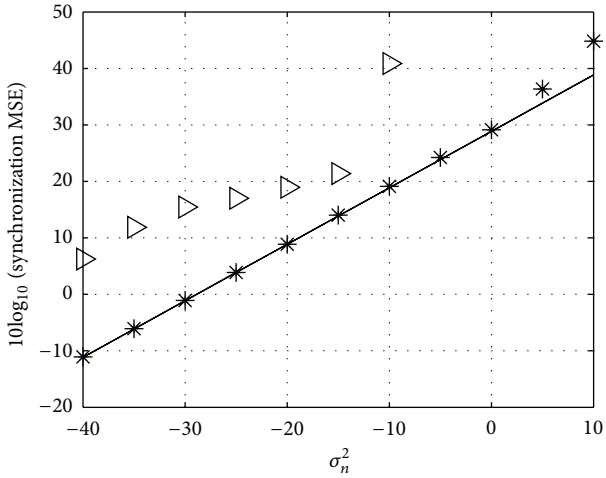


Figure 4: Synchronization accuracy for the source outside the sensor array. Solid line: tr(CRLB(τ)) from (6a), star symbol: synchronization MSE from the proposed method, right triangle symbol: synchronization MSE from the algorithm in [21].

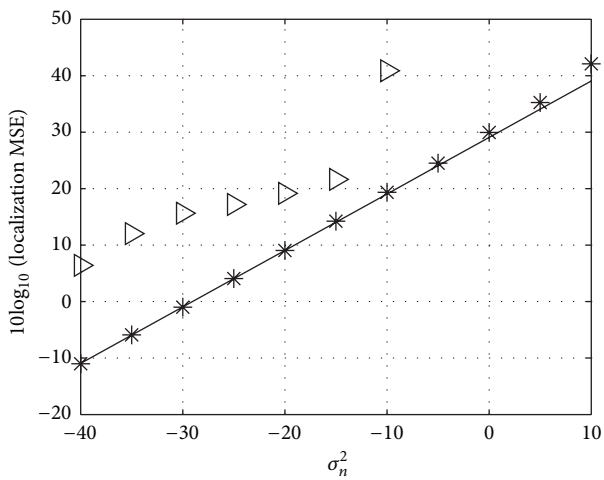


Figure 5: Localization accuracy for the source outside the sensor array. Solid line: tr(CRLB($\mathbf{p}$)) from (6b), star symbol: source localization MSE from the proposed method, right triangle symbol: source localization MSE from the algorithm in [21].

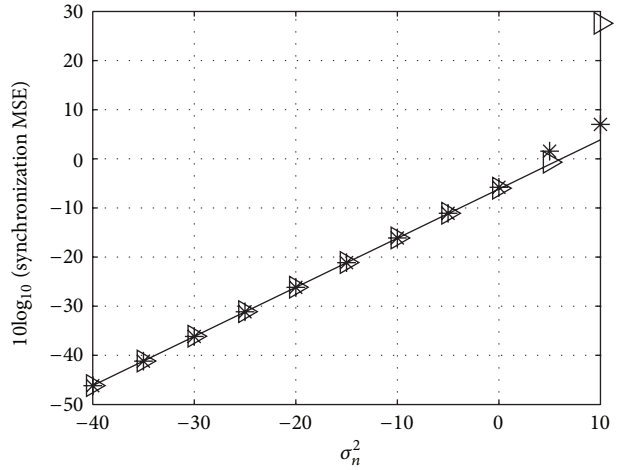


Figure 6: Synchronization accuracy for the source inside the sensor array. Solid line: tr(CRLB(τ)) from (6a), star symbol: synchronization MSE from the proposed method, right triangle symbol: synchronization MSE from the algorithm in [21].

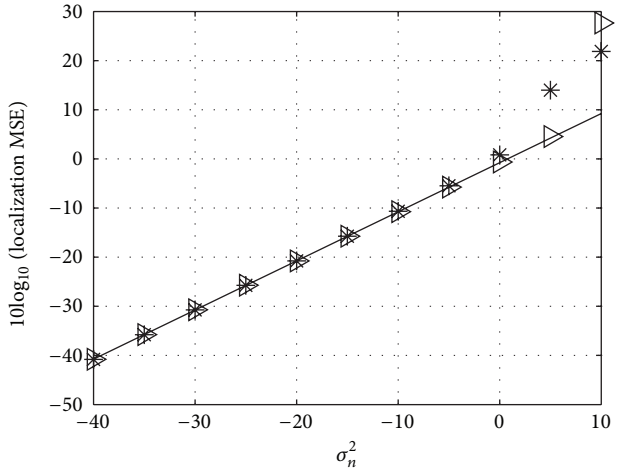


Figure 7: Localization accuracy for the source inside the sensor array. Solid line: tr(CRLB($\mathbf{p}$)) from (6b), star symbol: source localization MSE from the proposed method, right triangle symbol: source localization MSE from the algorithm in [21].

Figures 4 and 5 plot the clock bias and source position estimation MSEs for the source $\mathbf{p} = [50, 50]^T$ outside the sensor array. It can be seen from the figures that the proposed algorithm is able to attain the CRLB accuracy for the source position and clock bias before the TOA measurement noise power σ_n^2 reaches 0 in log scale. This is consistent with the theoretical performance analysis presented in Section 4.1 that the proposed algorithm is approximately efficient for accurate TOA measurements. On the other hand, the previously developed method from [21] cannot reach the CRLB accuracy, and its synchronization and localization MSEs are higher than the CRLB by an amount of more than 17 dB when σ_n^2 lies in the range from −40 to −25 in log scale. As the TOA measurement noise power further increases over −10 in log scale, the performance improvement due to the use of the newly proposed method is even more significant.

Figures 6 and 7 depict the estimation MSEs for the clock bias and the position of the source $\mathbf{p} = [-10, 10]^T$ inside the sensor array. Comparing with Figures 4 and 5 immediately reveals that the estimation performance is much better in this case, mainly because the localization geometry is improved. Moreover, in contrast to the case where the source is outside the sensor array, the proposed method and the algorithm from [21] yield similar estimation accuracy that matches the CRLBs for both the clock bias and the source position. This again verifies the performance analysis results in Section 4.1 on the approximate efficiency of the proposed solution. Interestingly, the method from [21] suffers from the threshold effect later than the new algorithm in this

simulation. However, its better performance does not persist as the source location moves outsides the sensor array, as shown in Figures 4 and 5. In contrast, the proposed algorithm can attain the CRLB performance for both cases where the source lies within and outside the sensor array, when the TOA measurement noise power is not sufficiently large.

6. Conclusions and Future Work

In this paper, the TOA-based joint synchronization and source localization problem was considered. Effects of neglecting the presence of the source clock bias in TOA measurements on source location estimation accuracy was first investigated. For this purpose, an MSE analysis was performed for the case where the source is localized via TOA positioning when assuming the source clock bias does not exist, but in fact it is nonzero. Comparing the obtained source localization MSE with that from joint estimating the source position and clock bias, we derived a condition under which ignoring the source clock bias may provide a smaller localization MSE. Numerical examples were provided to validate the analysis and reveal that, in some cases, neglecting the clock bias can severely degrade the localization performance. As a result, a new efficient closed-form solution for joint synchronization and source localization was proposed. The new method can identify the source location and clock bias using a twostep approach. Theoretical performance analysis and simulations were conducted to show that it can achieve the CRLB accuracy for both source location and clock bias estimates under small Gaussian TOA measurement noise.

Some recent work such as [22] adopted a more completed model where both the time offset (clock bias) and skew are considered. However, the proposed least squares (LS) estimator in [22] cannot reach the CRLB accuracy. As a future topic, we would like to derive an efficient closed-form estimator for joint synchronization and source localization in the presence of time skew.

Acknowledgments

The authors contributed equally to this paper. L. Yang's work is supported by the Startup Fund and the Youth Foundation of Jiangnan University (Contract no. JUSRP11234).

References

[1] R. J. Kozick and B. M. Sadler, "Source localization with distributed sensor arrays and partial spatial coherence," *IEEE Transactions on Signal Processing*, vol. 52, no. 3, pp. 601–616, 2004.

[2] S. Gezici, Z. Tian, G. B. Giannakis et al., "Localization via ultra-wideband radios: a look at positioning aspects of future sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 70–84, 2005.

[3] T. Li, A. Ekpenyong, and Y. F. Huang, "Source localization and tracking using distributed asynchronous sensors," *IEEE Transactions on Signal Processing*, vol. 54, pp. 3991–4003, 2006.

[4] H. C. So, "Source localization: algorithms and analysis," in *Handbook of Position Location: Theory, Practice and Advances*, S. A. Zekavat and R. M. Buehrer, Eds., chapter 2, pp. 25–66, Wiley-IEEE Press, Hoboken, NJ, USA, 2011.

[5] M. Z. Win, A. Conti, S. Mazuelas et al., "Network localization and navigation via cooperation," *IEEE Communications Magazine*, vol. 49, no. 5, pp. 56–62, 2011.

[6] K. W. Cheung, H. C. So, W. K. Ma, and Y. T. Chan, "Least squares algorithms for time-of-arrival-based mobile location," *IEEE Transactions on Signal Processing*, vol. 52, pp. 1121–1128, 2004.

[7] P. Biswas, T.-C. Liang, K.-C. Toh, T.-C. Wang, and Y. Ye, "Semidefinite programming approaches for sensor network localization with noisy distance measurements," *IEEE Transactions on Automation Science and Engineering*, vol. 3, pp. 360–371, 2006.

[8] F. Chan, H. C. So, and W. K. Ma, "A novel subspace approach for cooperative localization in wireless sensor networks using range measurements," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4548–4553, 2009.

[9] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proceedings of IEEE*, vol. 97, no. 2, pp. 427–450, 2009, Special issue on Ultra-Wide Bandwidth (UWB) technology and emerging applications.

[10] N. Patwari, A. O. Hero III, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2137–2148, 2003.

[11] K. W. Cheung and H. C. So, "A multidimensional scaling framework for mobile location using time-of-arrival measurements," *IEEE Transactions on Signal Processing*, vol. 53, no. 2, pp. 460–470, 2005.

[12] A. Beck, P. Stoica, and J. Li, "Exact and approximate solutions of source localization problems," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 1770–1778, 2008.

[13] M. Sun and K. C. Ho, "Successive and asymptotically efficient localization of sensor nodes in closed-form," *IEEE Transactions on Signal Processing*, vol. 57, no. 11, pp. 4522–4537, 2009.

[14] M. Sun, L. Yang, and K. C. Ho, "Accurate sequential self-localization of sensor nodes in closed-form," *Signal Processing*, vol. 92, no. 12, pp. 2940–2951, 2012.

[15] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts,"" in *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI '02)*, pp. 147–163, Boston, Mass, USA, December 2002.

[16] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 138–149, November 2003.

[17] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The flooding time synchronization protocol," in *Proceedings of the 8th International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 39–49, Baltimore, Md, USA, November 2004.

[18] K. Römer and F. Mattern, "Towards a unified view on space and time in sensor networks," *Computer Communications*, vol. 28, no. 13, pp. 1484–1497, 2005.

[19] B. Denis, J. B. Pierrot, and C. Abou-Rjeily, "Joint distributed synchronization and positioning in UWB Ad Hoc networks using TOA," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 4, pp. 1896–1910, 2006.

[20] H. Oliveira, E. Nakamura, and A. Loureiro, "Localization in time and space for sensor networks," in *Proceedings of the*

IEEE 21st International Conference on Advanced Information Networking and Applications (AINA '07), pp. 539–546, May 2007.

[21] S. Zhu and Z. Ding, "Joint synchronization and localization using TOAs: a linearization based WLS solution," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 1016–1025, 2010.

[22] J. Zheng and Y. C. Wu, "Joint time synchronization and localization of an unknown node in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1309–1320, 2010.

[23] S. Bancroft, "An algebraic solution of the GPS equations," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 21, no. 1, pp. 56–59, 1985.

[24] K. L. Noh, Q. M. Chaudhari, E. Serpedin, and B. W. Suter, "Novel clock phase offset and skew estimation using two-way timing message exchanges for wireless sensor networks," *IEEE Transactions on Communications*, vol. 55, no. 4, pp. 766–777, 2007.

[25] Z. Sahinoglu, "Improving range accuracy of ieee 802.15.4a radios in the presence of clock frequency offsets," *IEEE Communications Letters*, vol. 15, no. 2, pp. 244–246, 2011.

[26] Z. Sahinoglu and S. Gezici, "Enhanced position estimation via node cooperation," in *Proceedings of IEEE International Conference on Communications (ICC '10)*, pp. 23–27, May 2010.

[27] M. R. Gholami, S. Gezici, M. Rydstrom, and E. G. Strom, "A distributed positioning algorithm for cooperative active and passive sensors," in *Proceedings of the 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '10)*, pp. 1711–1716, September 2010.

[28] M. R. Gholami, S. Gezici, E. G. Strom, and M. Rydstrom, "Hybrid TW-TOA/TDOA positioning algorithms for cooperative wireless networks," in *Proceedings of IEEE International Conference on Communications (ICC '11)*, June 2011.

[29] B. Alavi and K. Pahlavan, "Modeling of the TOA-based distance measurement error using UWB indoor radio measurements," *IEEE Communications Letters*, vol. 10, no. 4, pp. 275–277, 2006.

[30] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Z. Win, "Ranging with ultrawide bandwidth signals in multipath environments," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 404–425, 2009, Special issue on Ultra-Wide Bandwidth (UWB) technology and emerging applications.

[31] A. Conti, M. Guerra, D. Dardari, N. Decarli, and M. Z. Win, "Network experiment for cooperative localization," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 2, pp. 467–475, 2012.

[32] Y. Shen and M. Z. Win, "Fundamental limits of wideband localization—part I: a general framwork," *IEEE Transactions on Information Theory*, vol. 56, no. 10, pp. 4956–4980, 2010.

[33] L. L. Scharf, *Statistical Signal Process*, Detection, Estimation and Time Series Analysis, Addison-Wesley, Reading, Mass, USA, 1991.

[34] Y. T. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Transactions on Signal Processing*, vol. 42, no. 8, pp. 1905–1915, 1994.

[35] S. M. Kay, *Fundamentals of Statistical Signal Processing, Estimation Theory*, Prentice Hall, Englewook Cliffs, NJ, USA, 1993.

[36] K. C. Ho, X. Lu, and L. Kovavisaruch, "Source localization using TDOA and FDOA measurements in the presence of receiver location errors: analysis and solution," *IEEE Transactions on Signal Processing*, vol. 55, no. 2, pp. 684–696, 2007.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 958912, 18 pages
http://dx.doi.org/10.1155/2013/958912

Research Article

A Novel Lightness Localization Algorithm Based on Anchor Nodes Equilateral Triangle Layout in WSNs

Dazhou Li, Hai Zhao, Jian Zhu, and Yuanguo Bi

College of Information Science and Engineering, Northeastern University, Shenyang, Liaoning 110819, China

Correspondence should be addressed to Dazhou Li; lidazhouzaku@gmail.com

Received 24 July 2012; Revised 13 November 2012; Accepted 10 December 2012

Academic Editor: Wei Meng

Localization, which determines the geographical locations of sensors, is a crucial issue in wireless sensor networks. In this paper, we propose a novel lightweight equilateral triangle localization algorithm (LETLA) that accurately localizes sensors and minimizes the power consumption. In the LETLA, the approximate coordinates substituted for the real coordinates of the unknown node, and the corresponding optimization problem is formulated to minimize the estimation error. With the sequences that represent the ranking of distances from the anchors to the unknown node, a simple and robust technique is developed to quickly and efficiently estimate a region containing the approximate coordinates, and a condition under which the approximate error can be minimized is given. This condition employs a new geometric construct of anchor layout called equilateral triangle diagrams. Extensive simulations show that the LETLA performs better than other state-of-the-art approaches in terms of energy consumption with the same localization precision.

1. Introduction

WSNs (Wireless Sensor Networks) have recently received great attention because they hold the potential to change many aspects of our economy and life [1–8]. Typical networks consist of a large number of densely deployed sensor nodes which could gather local data and communicate with other nodes. The data from these sensor nodes are relevant only if we know their locations. In addition, the accurate location estimation could aid in sensor network services such as routing, information processing, tasking, and querying. Therefore the knowledge of positions becomes imperative [9–17]. Moreover, the minimum resources must be used: typical sensor nodes are battery powered and have a limited processing ability [18, 19]. These constraints impose the new challenges in localization algorithm development and imply that power efficient, computation complexity and location precision should be employed simultaneously. Many methods have been proposed, such as APS (DV-Hop, DV-coordinates) [20], APIT [21–26], triangulation [27–29], Centroid [30–33], Sequence Based [34–38], Voronoi diagrams [39–43], and mathematical programming [44–46].

With regard to the precision of location, most of the localization algorithms can be classified into two broad categories: accurate location algorithm and approximate location algorithm. The accurate location algorithm produces the exact coordinates of the unknown node through complex calculations and precise measurement of distance. Typical accurate location algorithms include triangulation, linear programming, and semidefinite programming. The approximate location algorithms estimate an approximate position of the unknown node, with rough measurement and simple calculation. APS, APIT, Centroid, Sequence Based, and Voronoi diagrams are all approximate location algorithms.

The accurate location algorithms have two major requirements that render them disadvantages; that is, (a) the complexity is high, because triangulation, linear programming, and semidefinite programming usually involve solving higher order nonlinear equations which consume a large amount of energy, and (b) the ranging accuracy should be high enough, otherwise location algorithm will be halted. Under an adverse ranging condition, exact localization is not available; hence the statistical estimation will be introduced [47–49]. When the limited energy, the reduced processor, and the unstable

accuracy of ranging results have been given [5, 17, 19], the precise coordinates of the unknown node are usually difficult to be obtained. Furthermore, in the scene of mobile sensor nodes, the precise coordinates need so much time that the location result is not valid anymore.

The approximate location algorithms consume less energy in computing, since the location procedures of APS, APIT, Centroid, Sequence Based, and Voronoi diagrams are made up of simple logical operations and algebraic operations. The aim of approximate location algorithms is to find a rough precise position of the unknown node with the least energy consumption and to achieve a compromise between the accuracy and the complexity. The approximate location algorithms are particularly useful for large-scale wireless sensor networks, because they could extend sensor nodes' life with less computation complexity and energy consumption and tolerate ranging error to a certain degree.

In this paper, we present a novel equilateral triangle localization algorithm (LETLA) that is a lightweight approximate localization algorithm and could provide better precision with less power consumption. In the LETLA, the sensing area is covered by many identical disjoint equilateral triangle diagrams, and the anchors are placed in the vertexes of the equilateral triangle diagrams. The LETLA is an approximate localization based on the concept of substituting the approximate coordinates for the real coordinates, which could loose less accuracy and save more energy. The new geometric construct of the layout, called the equilateral triangle diagrams, has a major contribution to minimize the approximate error and simplify the location procedure. In order to avoid the ranging ambiguities arising from the interference of noise, the LETLA adopts the order of ranging results to represent the location relationship of unknown node and anchors. That is an effective technology and has been employed in many works of literature [34, 36–38].

The rest of the paper is organized as follows. The next section gives a brief overview of the related work. In Section 3, we describe the procedures of the LETLA. To prove the rationality of the LETLA, we calculate the utilization coefficient of the equilateral triangle diagrams, illustrate the geometrical characteristics of the equilateral triangle diagrams, explain the reason for dividing the equilateral triangle into seven distinct regions, and introduce a principle to determine the point of tangency. In Section 4, we demonstrate the localization procedures of the LETLA in a practical scenario. In Section 5, we present a performance study of the LETLA and make a comparison with other four localization techniques. We conclude this paper and mention our future work in Section 6.

2. State of the Art

In this section, we first give a brief summary of centroid, sequence-based, and Voronoi diagrams and then show the inspirations from them.

2.1. Survey of Centroid, Sequence-Based, and Voronoi Diagrams Localization Technology. Recently, many researchers have focused on localization in WSNs. Bulusu et al. [30] demonstrated a location technique called "Centroid" in 2000. Firstly, with the help of basic connectivity or distance information, a rough estimate of relative node distance is made. Then layout of anchors is used to create a relative map of anchor position. Finally the coordinates of the centroid are obtained by calculating the coordinates of anchors which surround the unknown node in a radiation range of communication. The coordinates of the centroid are regarded as the approximate coordinates of the unknown node. In 2007, weighted centroid localization (WCL) was provided by Blumenthal et al. [31]. It is derived from a centroid determination which calculates the position of unknown node by averaging the coordinates of anchors. To improve the precision in real implementations, the weights were used to refine the estimated position [32]. Because the radio device can provide the Link Quality Indication (LQI), received signal strength indication (RSSI), the performance of packet transmission, and even the difference of energy received in packets, it achieves better precision than original centroid localization. In 2011, Jun et al. [33] presented the first theoretical framework for WCL in terms of the localization error distribution parameterized by the node density, the node placement, the shadowing variance, the correlation distance, and the inaccuracy of sensor node positioning. With this analysis, the robustness of WCL has been quantified, and some design guidelines, such as node placement and spacing, for the practical deployment of WCL, have been provided.

Yedavalli et al. [34] described the sequence-based location approach, called Ecolocation, which was quite effective in dense and uniform topologies in 2005. Ecolocation determines the location of unknown nodes by examining the ordered sequence of received signal strength measurements taken at multiple reference nodes. The key features of the Ecolocation algorithm are as follows: (1) it constructs a constraint table based on the RSSI values; (2) it searches the table to find the location. However, the Ecolocation algorithm is imperfect and has a rough localization performance. In 2009, Zhong and He [50] presented a range-free approach to capture a relative distance between 1-hop neighboring nodes from their neighborhood orderings. With little overhead, the proposed method can be conveniently applied as a transparent supporting layer for many state-of-the-art connectivity-based localization solutions.

To improve the localization accuracy, a new algorithm based on the weighted rank order correlation coefficient and the dynamic centroid was proposed by Yu et al. [35] in 2011. The simulations indicate that the localization accuracy and the robustness of the new algorithm are distinctly raised compared with the Ecolocation.

In 2008, Yedavalli et al. [36] introduced a novel sequence-based localization technique (SBL). In the SBL, the localization space can be divided into many distinct regions that can be uniquely identified by the sequences. The sequences represent the ranking of distances from the anchors to that region. The SBL and the Ecolocation are both proposed by Kiran. The Ecolocation picks the location that maximizes the number of satisfied anchor's topology constraints. In contrast, the SBL applies two statistic metrics that capture

the difference in the rank orders. For the problem of location error, a new localization technique, based on the SBL, was proposed by Liu et al. [37] in 2009. Because it uses the triangular area which is enclosed by the centroids of the three "nearest" location regions, it improves the accuracy to some extent. In 2011, Hsiao et al. [38] analyzed the deployment strategy of sensor nodes for the SBL algorithm in order to effectively reduce location error such as (1) the standard deviation of the polygon area cut by the perpendicular bisectors should be kept as small as possible; (2) certain amount of space should be maintained between the sensor nodes, and (3) optimization with the angle between the perpendicular bisectors should be utilized.

Voronoi diagrams provide a powerful technique for analysing computational geometric problems. The Voronoi diagrams divide the plane into multiple polygons (known as cells). Specifically, the cells are constructed in the way that any point in a cell is closer to the local site (i.e., the site within the cell) than to any other site on the plane. Recently, the concept of Voronoi diagrams has been applied in robot navigation and map establishing [39].

In 2007, Boukerche et al. [40] proposed a novel approach that adopted Voronoi diagrams. Two types of localization can result from the proposed algorithm: the physical location of the node (e.g., latitude, longitude) or a region limited by the node's Voronoi cell. In 2009, Boukerche et al. improved DV-Hop localization algorithm with Voronoi diagrams to limit the scope of flooding communication and error [41]. In 2010, to achieve "k-coverage" of the sensing area, which means every point in the surveillance area is monitored by at least k sensors, Li and kao [42] presented a novel distributed self-location estimation scheme based on Voronoi diagrams with mobile nodes. Li illustrated that distributed Voronoi diagrams provided a convenient means of analysing the coverage problem in large-scale sensor networks. In the same year, Ampeliotis and Berberidis [43] generalized a notion of the closest point of the approach estimator. While in the closest point of approach estimator, the unknown node may lie close to the closest point of approach node. Hence the unknown node is restricted to lie in a convex set called the sorted order-K Voronoi cell.

2.2. Inspiration of Centroid, Sequence-Based, and Voronoi Diagrams Localization Technology. By the literature in former section, we find some inspirations from centroid, sequence-based, and Voronoi diagrams. They are adopted, developed, and fused in the LETLA.

According to the facts in WSNs, most location applications only require a proximate region instead of the accurate coordinates of the unknown nodes. Especially in the localization of the mobile nodes, the real-time and approximate coordinates of the mobile nodes are more effective than the accurate coordinates with a long computing delay. Moreover, WSNs need to prolong the sensor node's life, as the sensor nodes are powered by batteries and replaced difficultly. Therefore, substituting the approximate coordinates for the real coordinates of the unknown node is an efficient method to balance the location precision and the computation complexity. The purpose of centroid is to substitute the approximate coordinates for the real coordinates, which significantly reduces the location complexity. Enlightened by the centroid, we proposed a method to obtain the approximate coordinates instead of the real coordinates. This method simplifies the arc equations to the tangent equations as described in Section 3.

In WSNs, the unknown nodes can build Voronoi diagrams based on the position information of anchors and the rank of ranging result. Each node also can find the Voronoi diagrams it belongs to. There is a geometric constraint that links the location to the sorting of the distances between the unknown node and the anchors. If we know the correct sorting of the distances, we would restrict the space in which the unknown node may lie. This space is the Voronoi diagrams that corresponds to the correct sorting. Aiming to reduce the computation complexity in building Voronoi diagrams, we propose the equilateral triangle diagrams which have better geometric attribute than Voronoi diagrams in simplifying location process.

The order sequences of ranging result are more robust than the numerical measurement of distance, which has been proved in some papers [34, 36–38]. The measurement noise corrupts the numerical measurement of distance directly and distinctly over the whole location area, but it alters the rank of ranging result slightly in the most of the areas. In the Voronoi diagrams and the equilateral triangle diagrams, each region can be identified by only a sorted sequence of ranging result. Hence, the special limited region covering the unknown node can be determined by the order sequences of ranging result. In some mobile instances, the special limited region can be regarded as a rough location, if a little location delay is demanded. Consequently we apply the sequence-based method in the LETLA to determine the special limited region efficiently.

3. Lightness Equilateral Triangle Localization Method

In the LETLA, the anchors are able to acquire their positions via external device like GPS, but the unknown nodes cannot obtain this information. The unknown nodes can get the positions of the anchors and construct the corresponding location sequence tables. Then, every wireless sensor node is equipped with an omnidirectional antenna which can transfer wireless signal in all directions [21, 50–56].

The localization area is divided into many same equilateral triangles, as shown in Figure 1(a). Each equilateral triangle formed by three anchors is picked out to estimate the position of the unknown node. All anchors are deployed in the vertexes of equilateral triangles and fixed after the initial deployment. Each anchor has six adjacent anchors with the same distance, denoted by a. As shown in Figure 1(a), anchor J has six adjacent anchors K, L, M, N, R, and S, which are the nearest six anchors to J with the distance of a. Each node has the same data transmission radius, denoted by l. In the LETLA $l > 2a$ is valid. $\triangle$ denotes an equilateral triangle. The basic procedures of the LETLA can be described as follows.

Step 1. The unknown node measures the distances between itself and the anchors in its transmission range and sorts

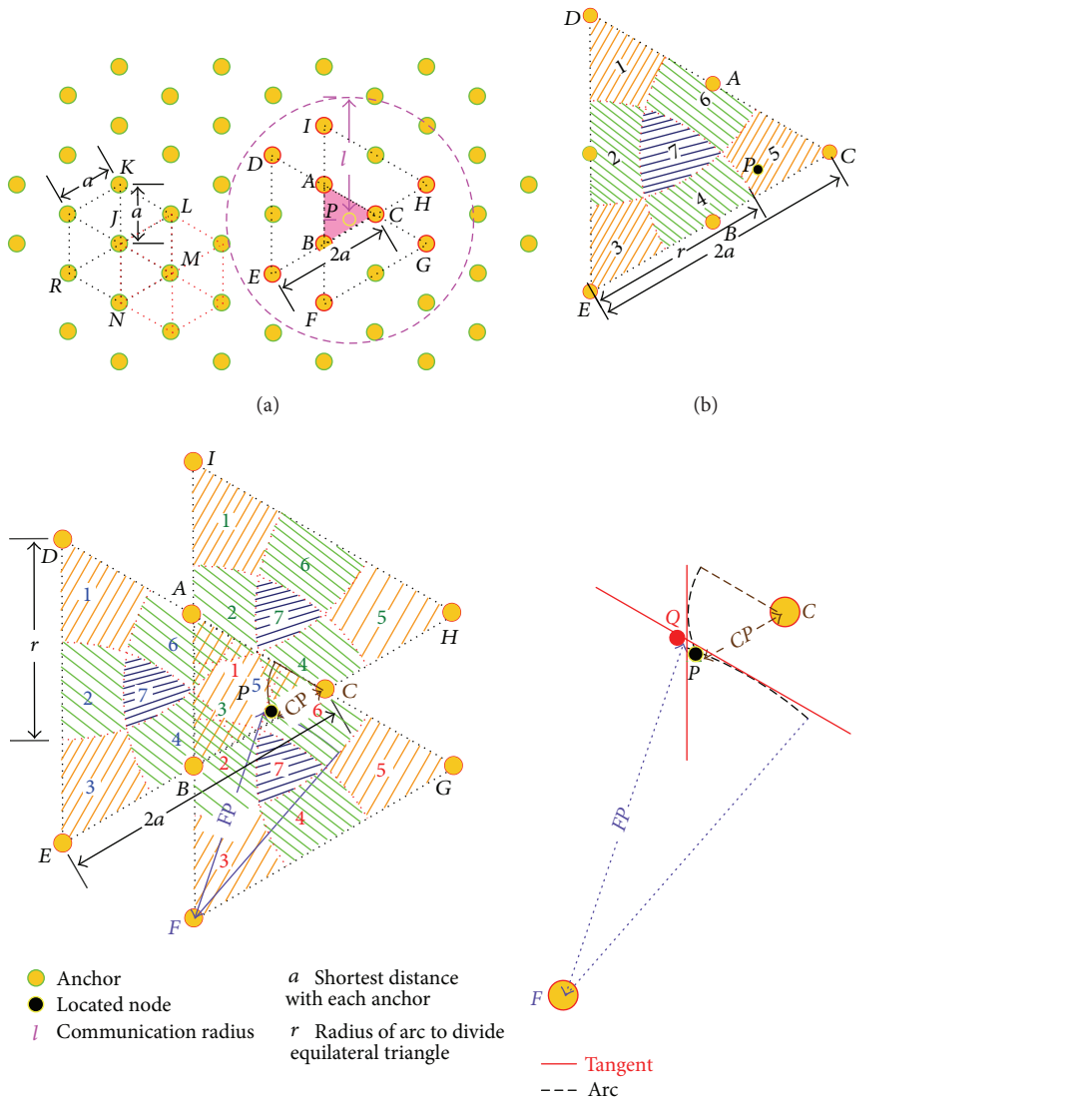


FIGURE 1: (a) The anchor layout and a demonstration of Steps 1 and 2; (b) the example of dividing $\triangle DEC$ in Step 3; (c) an instance of selecting two arcs in Step 4; (d) the intersection of two tangents substituted for the intersection of two arcs in Step 5.

the ranging result in the ascending order. The node P is an unknown node and resides in an equilateral triangle with the side of a. This equilateral triangle can be found by utilizing the three shortest distances between the anchors and node P. If nodes A, B, and C are the three nearest anchors according to the ranging sequence, the node P will reside in the $\triangle ABC$, as shown in Figure 1(a). In Step 1, node P is assigned to $\triangle ABC$, only by means of the rank of distance measurement, not related to any numerical computation.

Step 2. When $\triangle ABC$ is determined, $\triangle DEC$, $\triangle IBH$, and $\triangle AFG$ are selected. As shown in Figure 1(a) the overlap region of $\triangle DEC$, $\triangle IBH$, and $\triangle AFG$ is $\triangle ABC$ and their side lengths are $2a$. Because node P knows the positions of the anchors, it is feasible to find these three equilateral triangles.

Step 3. $\triangle DEC$, $\triangle IBH$, and $\triangle AFG$ are divided into seven distinct regions, respectively. Figure 1(b) shows the example of dividing $\triangle DEC$ into seven distinct regions, by means of three arcs with the centers on the vertexes of the $\triangle DEC$ and the radii equal to r. Node P localizes itself at the fifth region of the $\triangle DEC$ according to the relation of DP, EP, CP, and r in Table 1. Node P also finds other two regions in $\triangle IBH$ and $\triangle AFG$, respectively, in the same way.

Step 4. According to the regions including node P in $\triangle DEC$, $\triangle IBH$, and $\triangle AFG$, two arcs are selected. They have an intersection in the position of node P. For the example presented in Step 3, two arcs are shown in Figure 1(c). Because node P is in the fifth region of $\triangle DEC$, the third region of $\triangle IBH$, and the sixth region of $\triangle AFG$ in Step 3, the arc with

Table 1: The method of determining the sequence number of the region in $\triangle DEC$.

Sequence number	Sufficient and necessary condition
1	$DP < r, EP > r, CP > r$
2	$DP < r, EP < r, CP > r$
3	$DP > r, EP < r, CP > r$
4	$DP > r, EP < r, CP < r$
5	$DP > r, EP > r, CP < r$
6	$DP < r, EP > r, CP < r$
7	$DP < r, EP < r, CP < r$

center F and radius FP and the arc with center C and radius CP are selected by Table 2. The length of the arc is limited by the range of region divided in Step 3. As shown in Figure 1(c), because node P is in the sixth region of $\triangle AFG$, the length of one arc is limited by the range of the sixth region in $\triangle AFG$. In the same manner, length of the other arc is limited by the range of the fifth region in $\triangle DEC$.

Step 5. The intersection of the two tangents is substituted for the intersection of two arcs selected in Step 4. In Figure 1(d), the intersection of the two tangents, denoted by Q, is considered to be the approximation of the intersection of two arcs which is also the position of the unknown node P. The principle of finding the point of the tangency is illustrated in Section 3.3.

3.1. Geometrical Characteristics of Equilateral Triangle Diagram. In Step 1, we propose a proposition: the unknown node P resides in $\triangle ABC$, when nodes A, B, and C are the three nearest anchors in Figure 1(a). In this section, we validate the proposition through mathematical proofs. We consider the sides of an equilateral triangle as a part of the inside of the equilateral triangle. Thus, a node is either inside or outside an equilateral triangle.

Proposition 1. *As shown in Figure 1(a), if nodes A, B, and C are the three nearest anchors to node P according to the ranging sequences, node P is inside $\triangle ABC$.*

Proof. First, in Figure 1(a), the localization space is cut into many equilateral triangles with the same side length a, and therefore each node in the localization space will be inside an equilateral triangle with the side length a. Second, if a node is inside an equilateral triangle, the distances between the node and the three vertexes of the equilateral triangle are all less than the side length (a) of the equilateral triangle. If a node is outside an equilateral triangle, the longest distance between the node and the three vertexes of the equilateral triangle is more than a.

Let us prove this by contradiction. Nodes A, B, and C are the three nearest anchors to node P according to the ranging sequences. Let us suppose the false proposition that node P is outside $\triangle ABC$. Since each node in the localization space will be inside an equilateral triangle with the side length a, node P is inside $\triangle JKL$, which is an arbitrary equilateral triangle and differs from the $\triangle ABC$. Because node P is inside $\triangle JKL$, it can be concluded that $PJ < a$, $PK < a$ and $PL < a$. Node P is outside $\triangle ABC$ simultaneously, if $PA > PB > PC$, hence it is obtained $PA > a$. Consequently, we can draw a conclusion that $PA > PJ$, $PA > PK$, and $PA > PL$, which is a contradiction as nodes A, B, and C, are the three nearest anchors to node P. Now it is fallacious that node P is outside $\triangle ABC$. So the proposition that node P is inside $\triangle ABC$ is true. □

3.2. Reason of Dividing Equilateral Triangle. In this section, we will present the reason why $\triangle DEC$, $\triangle IBH$, and $\triangle AFG$ are divided into seven distinct regions, respectively. We first introduce a conclusion which will be proved in Section 3.3. In substituting the intersection of the two tangents for the intersection of the two arcs, as described in Step 5, the approximation error varies directly with the angle and the radius of the two arcs. For a given point of tangency, less radius and angle of the arc will lessen the approximation error.

Take $\triangle DEC$ for example, only node D, node E, and node C can be considered as the center of the arc in Table 2. Before dividing, the angle of the arc in $\triangle DEC$ is 60 degrees and the radius of the arc in $\triangle DEC$ is in the range $[0, a]$. When $\triangle DEC$ has been divided into seven distinct regions, the length of the arc is limited by the range of the region which reduces the ranges of angle and radius of the arc in $\triangle DEC$. The arcs in the first, third, and fifth regions have less radii which is about a third of the original length, and the arcs in the second, fourth, sixth, and seventh regions have less angles about 30 degrees. We can draw a conclusion that dividing $\triangle DEC$ into seven regions effectively reduces the range of the angle and the radius of the arc in $\triangle DEC$. It will be helpful to lessen the approximate error in Step 5.

As shown in Figure 1(a), $\triangle ABC$ is the overlap region of $\triangle DEC$, $\triangle IBH$, and $\triangle AFG$; thereby $\triangle ABC$ is made up of ten divisions which are superposed by different regions of $\triangle DEC$, $\triangle IBH$, and $\triangle AFG$, respectively. The ten divisions in $\triangle ABC$ are denoted by α, β, σ, χ, δ, ε, γ, φ, θ, and μ, as demonstrated in Figure 2(b). The definitions of the ten divisions are according the superposition of the region in $\triangle DEC$, $\triangle IBH$, and $\triangle AFG$, as presented in Table 3. By the comparison between Tables 2 and 3, the selection of center and radius of the arc in Step 4 is based upon the division including the unknown node P.

The length of radius in Step 3, denoted by r, is a function of a. Given a, r is presented according to the cosine theorem of sides in Figure 2(c):

$$r^2 = (2a)^2 + \left(\frac{\sqrt{3}}{2}a\right)^2 - 2 \times 2a \times \frac{\sqrt{3}}{2}a \times \cos\left(\frac{\pi}{6}\right), \quad (1)$$

$$r = \frac{\sqrt{7}}{2}a. \quad (2)$$

3.3. Principle of Determining Point of Tangency. An arc is determined by the coordinates of center, the length of radius,

Table 2: The method of selecting two arcs to indicate unknown node P with their intersection.

Sequence number in $\triangle DEC$	Sequence number in $\triangle IBH$	Sequence number in $\triangle AFG$	Center	Radius
6	2	1	Node A and node H	AP and HP
4	3	2	Node B and node D	BP and DP
5	4	6	Node C and node F	CP and FP
6	3	1	Node A and node E	AP and EP
5	2	1	Node A and node H	AP and HP
4	3	1	Node B and node D	BP and CP
5	3	2	Node B and node G	BP and GP
5	3	6	Node C and node F	CP and FP
5	4	1	Node C and node I	CP and IP
5	3	1	Node A and node B	AP and BP

Table 3: The definition of the ten divisions in $\triangle ABC$.

Sequence number in $\triangle DEC$	Sequence number in $\triangle IBH$	Sequence number in $\triangle AFG$	Symbol in $\triangle ABC$
6	2	1	α
4	3	2	δ
5	4	6	γ
6	3	1	β
5	2	1	σ
4	3	1	χ
5	3	2	ε
5	3	6	φ
5	4	1	θ
5	3	1	μ

and the value of angle. In a given arc, a tangent is determined by the position of the point of tangency. Hence, the selection of the tangent is equivalent to choosing an optimization position of the point of tangency. The tangent which touches the arc in the optimal point can minimize the approximation error in Step 4. The gap between the arc and the tangent is a major effect factor of approximation error in Step 5. To facilitate the calculation, a segment of the tangent is used to substitute the whole tangent. As shown in Figure 3, FE is a segment of the tangent and intercepts by two lines parallel with the bisector of the arc. AB is a given arc with a given radius and a given value of degree, denoted by r and a, respectively. The origin of the coordinate plane is denoted by O, which is located at the center of AB, and the y-axis is the bisector of AB and intersects AB at C. D is the point of tangency and b denotes the degree between the side of AB and the radius perpendicular to FE. D can be any point on AB, so the gap between the arc and the tangent is affected by the position of D.

The effects of substituting EF for AB can be evaluated $\text{Error}_{\text{gap}}$. $(x_{\text{tan}}, y_{\text{tan}})$ denotes the coordinates of a point on EF and $(x_{\text{arc}}, y_{\text{arc}})$ denotes the coordinates of a point on AB. The definition of $\text{Error}_{\text{gap}}$ is the difference between y_{tan} and y_{arc} when x_{tan} is equal to x_{arc}. In Figure 3, a double-headed arrow in the gap between EF and AB is an $\text{Error}_{\text{gap}}$:

$$\text{Error}_{\text{gap}} = y_{\text{tan}} - y_{\text{arc}}, \quad \left(x_{\text{tan}} = x_{\text{arc}} = r \times \cos\left(\frac{\pi - a}{2} + c\right)\right). \tag{3}$$

The range of both x_{tan} and y_{tan} is (4). Because each value of the x-axis in (4) corresponds to an $\text{Error}_{\text{gap}}$, the number of $\text{Error}_{\text{gap}}$ approaches infinity:

$$r \times \cos\left(\frac{\pi - a}{2}\right) \le x_{\text{tan}} \le r \times \cos\left(\frac{\pi + a}{2}\right), \quad r \times \cos\left(\frac{\pi - a}{2}\right) \le x_{\text{arc}} \le r \times \cos\left(\frac{\pi + a}{2}\right). \tag{4}$$

As $(x_{\text{tan}}, y_{\text{tan}})$ denotes the coordinates of a point on EF and $(x_{\text{arc}}, y_{\text{arc}})$ denotes the coordinates of a point on AB, the expressions of y_{tan} and y_{arc} can be obtained with the angle a, b, and c.

$$y_{\text{tan}} = r \times \tan\left(\pi - \frac{a}{2} + b\right) \times \left[\sin\left(\frac{a}{2} - c\right) - \sin\left(\frac{a}{2} - b\right)\right] + r \times \cos\left(\frac{a}{2} - b\right), \quad y_{\text{arc}} = r \times \sin\left(\frac{\pi}{2} - \frac{a}{2} + c\right) = r \times \cos\left(\frac{a}{2} - c\right). \tag{5}$$

By combining (3) and (5), the following new expression of $\text{Error}_{\text{gap}}$ can be obtained in (6). Specifically, $\text{Error}_{\text{gap}}$ is the function of r, a, b, and c and is actually in proportion to r and a according to (6). When AB is fixed, r and a are given. The parameter b is determined by the position of D. The parameter c in (6) has the range of $[0, a]$. Finding the optimal position of D is equivalent to finding the optimal value of angle b to minimize the $\text{Error}_{\text{gap}}$ over the whole feasible range of angle c, in the given radius r and angle a. Three indicators can quantify the effect of $\text{Error}_{\text{gap}}$ over the whole gap between

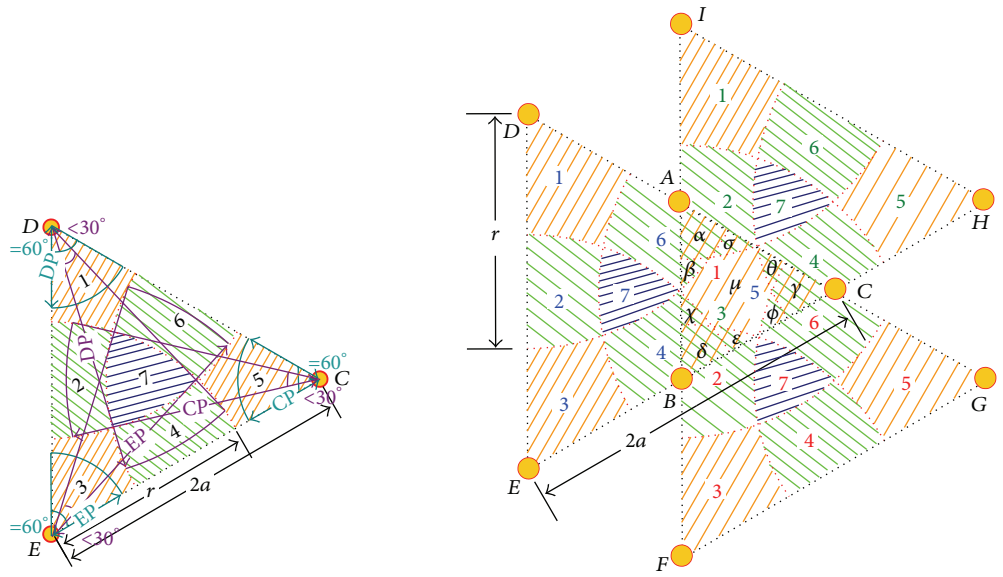


FIGURE 2: (a) The angle and the radius of the arc in different regions of $\triangle DEC$ after dividing; (b) the ten divisions in $\triangle ABC$; (c) applying the cosine theorem of sides to obtain r.

the EF and AB and are the functions of angle b:

$$\text{Error}_{\text{gap}} = r \times \left(\frac{1 - \sin((a/2) - b) \times \sin((a/2) - c)}{\cos((a/2) - b)} - \frac{\cos((a/2) - b) \times \cos((a/2) - c)}{\cos((a/2) - b)} \right) = r \times \left(\frac{1 - \cos(b - c)}{\cos((a/2) - b)} \right). \quad (6)$$

The first indicator is $E(\text{Error}_{\text{gap}})$ which is defined as the average of $\text{Error}_{\text{gap}}$ over the whole gap between EF and AB.

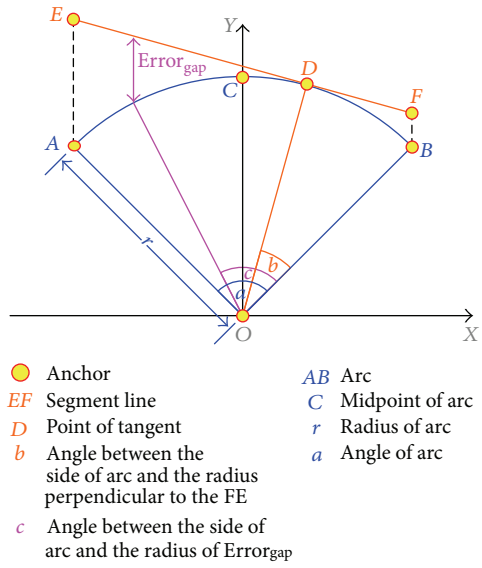


FIGURE 3: The definition of Error$_{gap}$.

x_A and x_B denote the coordinates of A and B in the x-axis and $[x_A, x_B]$ represents the whole feasible range on the x-axis over the whole gap between EF and AB. Then the expression of $E(\text{Error}_{gap})$ is obtained:

$$E\left(\text{Error}_{gap}\right) = \frac{1}{x_B - x_A}\int_{x_A}^{x_B} \text{Error}_{gap}\,dx. \tag{7}$$

$[x_A, x_B]$ in the x-axis can be represented by $[0, a]$ in angle c. By differential calculus of (3), we obtain

$$dx = -r \times \sin\left(\frac{\pi}{2} - \frac{a}{2} + c\right) \times dc. \tag{8}$$

By combining (7) and (8),

$$\begin{aligned} E\left(\text{Error}_{gap}\right) &= \frac{1}{x_B - x_A}\int_{x_A}^{x_B} \text{Error}_{gap} \times dx \\ &= \frac{1}{2\sin(a/2)}\int_0^a \text{Error}_{gap} \times \cos\left(\frac{a}{2} - c\right) \times dc. \end{aligned} \tag{9}$$

By combining (6) and (9), the new expression of $E(\text{Error}_{gap})$ is given by

$$\begin{aligned} E\left(\text{Error}_{gap}\right) = {} & \frac{r}{2\sin(a/2)} \\ & \times \left[\frac{8\sin(a/2) - \sin((3/2)a - b)}{4\cos((a/2) - b)}\right. \\ & \left. - \frac{\sin((a/2) + b)}{4\cos((a/2) - b)} - \frac{a}{2}\right]. \end{aligned} \tag{10}$$

Because AB is given, radius r and angle a are fixed. The value of $E(\text{Error}_{gap})$ only accounts for the value of angle b.

The extremum problem can be solved by finding the value of angle b which makes the first derivative of $E(\text{Error}_{gap})$ equal to zero:

(i) syms ab,

(ii) solve(diff('$(8 * \sin(0.5 * a) - \sin(1.5 * a - b) - \sin(0.5 * a + b))/(4 * \cos(0.5 * a - b))$','$b$'),'$b$').

The numerical result given by MATLAB demonstrates that $E(\text{Error}_{gap})$ can be least only if b is equal to $0.5a$. It shows that $b = 0.5a$ is the condition for minimizing $E(\text{Error}_{gap})$ and means that the point of tangency is the midpoint of the arc. When D is the midpoint of AB in Figure 3, the average of Error$_{gap}$ over the whole gap between EF and AB is minimal.

The second indicator is $D(\text{Error}_{gap})$ which is defined as the variance of Error$_{gap}$ over the whole gap between EF and AB. $[x_A, x_B]$ represents the whole feasible range on the x-axis over the whole gap between EF and AB. Then the expression of $D(\text{Error}_{gap})$ is obtained:

$$D\left(\text{Error}_{gap}\right) = \frac{1}{x_B - x_A}\int_{x_A}^{x_B}\left(\text{Error}_{gap} - E\left(\text{Error}_{gap}\right)\right)^2 dx. \tag{11}$$

By applying (8) to (11), (12) is obtained:

$$\begin{aligned} D\left(\text{Error}_{gap}\right) &= \frac{1}{x_B - x_A}\int_{x_A}^{x_B}\left(\text{Error}_{gap} - E\left(\text{Error}_{gap}\right)\right)^2 \\ &\quad \times dx \\ &= \frac{1}{2\sin(a/2)}\int_0^a \left(\text{Error}_{gap} - E\left(\text{Error}_{gap}\right)\right)^2 \\ &\quad \times \cos((a/2) - c) \times dc. \end{aligned} \tag{12}$$

Combined with (6), (12) can be changed to (13). It is the new expression of $D(\text{Error}_{gap})$:

$$\begin{aligned} D\left(\text{Error}_{gap}\right) = {} & \left(\frac{r}{\cos((a/2) - b)} - E\left(\text{Error}_{gap}\right)\right)^2 \\ & + r \times \left(\frac{\cos((a/2) - b) \times E\left(\text{Error}_{gap}\right) - r}{\cos^2((a/2) - b)}\right) \\ & \times \left(\frac{2\cos((a/2) - b) + \sin((3/2)a - b)}{4\sin(a/2)}\right. \\ & \left. + \frac{\sin((a/2) + b)}{4\sin(a/2)}\right) \\ & + r^2\left(\frac{3\sin((3/2)a - 2b)}{24\sin(a/2)\cos^2(a/2 - b)}\right. \\ & \left. + \frac{\sin((5/2)a - 2b)}{24\sin(a/2)\cos^2(a/2 - b)}k\right. \end{aligned}$$

$$- \frac{3\sin(a/2 - 2b)}{24\sin(a/2)\cos^2(a/2 - b)} + \frac{\sin(a/2 + b) + 12\sin(a/2)}{24\sin(a/2)\cos^2((a/2) - b)}\Bigg). \tag{13}$$

Because AB is given, radius r and angle a are fixed. $D(\text{Error}_{\text{gap}})$ only accounts for the value of angle b. The numerical result given by MATLAB demonstrates that $D(\text{Error}_{\text{gap}})$ can be least only if b is equal to $0.5a$. It shows that $b = 0.5a$ is the condition for minimizing $D(\text{Error}_{\text{gap}})$ and means that the point of tangency is the midpoint of the arc. When D is the midpoint of AB in Figure 3, the variance of $\text{Error}_{\text{gap}}$ over the whole gap between EF and AB is minimal.

(i) syms $a\,b\,s1\,s2\,s3\,s4\,s5$;

(ii) $s1 = (1/(2 * \sin(0.5 * a))) * (((8 * \sin(0.5 * a) + \sin(b - 1.5 * a) - \sin(b + 0.5 * a))/(4 * \cos(b - 0.5 * a))) - (0.5 * a))$;

(iii) $s2 = 2 * \sin(0.5 * a) * (1/(\cos(b - 0.5 * a)^2\ )) + (s1)^2 - (2 * (s1))/(\cos(b - 0.5 * a)))$;

(iv) $s3 = ((2 * (s1))/(\cos(b - 0.5 * a)) - (2)/((\cos(b - 0.5 * a))^2)) * ((0.5 * \cos(b - 0.5 * a)) - (0.25 * \sin(b - 1.5 * a)) + (0.25 * \sin(b + 0.5 * a)))$;

(v) $s4 = (1/(2 * (\cos(b - 0.5 * a)^2))) * ((0.5 * \sin(1.5 * a - 2 * b)) - ((1/6) * (\sin(2 * b - 2.5 * a))) - (0.5 * \sin(0.5 * a - 2 * b)) + ((1/6) * (\sin(2 * b + 0.5 * a))) + (2 * \sin(0.5 * a)))$;

(vi) $s5 = s2 + s3 + s4$;

(vii) solve(diff($s5,'b'$),$'b'$).

The third indicator is $\text{MAX}(\text{Error}_{\text{gap}})$ which is defined as the maximal value of $\text{Error}_{\text{gap}}$ over the whole gap between EF and AB, given a fixed position of$D$, when the position of$D$ is fixed, which means that the degree of angle b is constant. Considering the definition of $\text{Error}_{\text{gap}}$, the maximal value of $\text{Error}_{\text{gap}}$ can be found in $c = 0$ or $c = a$ and depends on the degree of angle b. Thus, the expression of $\text{MAX}(\text{Error}_{\text{gap}})$ is obtained:

$$\text{Max}\left(\text{Error}_{\text{gap}}\right) = \begin{cases} r \times \left(\dfrac{1 - \cos(a - b)}{\cos((a/2) - b)}\right), \\ \qquad 0 \le b < \dfrac{a}{2},\ c = a \\ r \times \left(\dfrac{1 - \cos(a/2)}{1}\right), \\ \qquad b = \dfrac{a}{2},\ c = 0 \text{ or } c = a \\ r \times \left(\dfrac{1 - \cos(b)}{\cos((a/2) - b)}\right), \\ \qquad \dfrac{a}{2} \le b < a,\ \text{c} = 0. \end{cases} \tag{14}$$

Proposition 2. *As shown in Figure 3 and indicated in* (14)*. If the degree of angle a is constant and the degree of angle b is in the interval* $[0, 0.5a)$, $\text{Max}(\text{Error}_{\text{gap}})$ *will be reached in* $c = a$ *when* $b = 0.5a$.

Proposition 3. *As shown in Figure 3 and indicated in* (14)*, if the degree of angle a is constant and the degree of angle b is in the interval* $(0.5a, a]$, $\text{Max}(\text{Error}_{\text{gap}})$ *will be reached in* $c = 0$ *when* $b = 0.5a$.

Proof. Consider

$\because 0 \le b < a/2$ and $0 < a < \pi$

$\therefore -a/2 < -b \le 0$

$\therefore a/2 < a - b \le a$

$\because 0 < a/2 < \pi/2$

$\therefore 1 - \cos(a/2) < 1 - \cos(a - b)$

$\because 1 > \cos(a/2 - b)$

$\therefore\ r \times ((1 - \cos(a/2))/1) < r \times ((1 - \cos(a - b))/\cos((a/2) - b))$.

□

Proof. Consider

$\because a/2 < b \le a$ and $0 < a < \pi$

$\therefore 0 < a/2 < \pi/2$

$\therefore 1 - \cos(a/2) < 1 - \cos(b)$

$\because 1 > \cos(a/2 - b)$

$\therefore r \times ((1 - \cos(a/2))/1) < r \times ((1 - \cos(b))/\cos((a/2) - b))$.

□

Therefore, if a is given, we can obtain the minimal $\text{Max}(\text{Error}_{\text{gap}})$ in $b = 0.5a$. It means that $b = 0.5a$ is the necessary and sufficient condition for minimizing $\text{Max}(\text{Error}_{\text{gap}})$. Taking $\text{Max}(\text{Error}_{\text{gap}})$ into consideration, consequently, the point of the tangent must be the midpoint of the arc. When D is the midpoint of AB in Figure 3, the maximal value of $\text{Error}_{\text{gap}}$ over the whole gap between EF and AB is less than other positions of D.

Given an arc, the tangent is dominated by the position of the point of tangency. If the point of tangency is the midpoint of the arc, then $E(\text{Error}_{\text{gap}})$, $D(\text{Error}_{\text{gap}})$, and $\text{Max}(\text{Error}_{\text{gap}})$ have the least value in each range. It has been used in the LETLA to choose the point of tangency.

3.4. Utilization Coefficient of Equilateral Triangle Diagrams. In Step 4, two anchors are selected from the nine anchors which have the special geometrical relations as shown in Figure 1(a). However, it is hard to satisfy the geometrical requirement in the boundaries of the location space. In Figure 4, the location area, unlocation area, and cover area are defined and illustrated in three examples with the sides $(3a, a)$, $(4a, 2a)$, and $(5a, 3a)$ of the cover area. In this section, the utilization coefficient of equilateral triangle diagrams is obtained, which means the percentage of the location area in the cover area. As the sides of the cover area increase linearly, the number of anchors and the location area, the unlocation area, and the cover area are listed in Table 4.

$(m \times a, n \times a)$ denotes the length of the sides of the cover area in Figure 4 the formulas of $\text{Num}_{\text{anchor}}$, S_{unlocat}, S_{locat}, and S_{cover} are obtained as follows. $\text{Num}_{\text{anchor}}$ is the number of anchors and $\text{S}_{\text{unlocat}}$ is the unlocation area. Because $m - n = 2a$ is the relation between m and n in terms of Table 4, (15), (16),

Table 4: The relation of the five indexes.

The side length of cover area	$(3a, 1a)$	$(4a, 2a)$	$(5a, 3a)$	$(6a, 4a)$	$(7a, 5a)$	$(8a, 6a)$
The number of anchors	18	36	60	90	126	168
Unlocation area	$18 \times \frac{a^2\sqrt{3}}{4}$	$30 \times \frac{a^2\sqrt{3}}{4}$	$42 \times \frac{a^2\sqrt{3}}{4}$	$54 \times \frac{a^2\sqrt{3}}{4}$	$66 \times \frac{a^2\sqrt{3}}{4}$	$78 \times \frac{a^2\sqrt{3}}{4}$
Location area	$4 \times \frac{a^2\sqrt{3}}{4}$	$22 \times \frac{a^2\sqrt{3}}{4}$	$52 \times \frac{a^2\sqrt{3}}{4}$	$94 \times \frac{a^2\sqrt{3}}{4}$	$148 \times \frac{a^2\sqrt{3}}{4}$	$214 \times \frac{a^2\sqrt{3}}{4}$
Cover area	$22 \times \frac{a^2\sqrt{3}}{4}$	$52 \times \frac{a^2\sqrt{3}}{4}$	$94 \times \frac{a^2\sqrt{3}}{4}$	$148 \times \frac{a^2\sqrt{3}}{4}$	$214 \times \frac{a^2\sqrt{3}}{4}$	$292 \times \frac{a^2\sqrt{3}}{4}$

(17), and (18) can be simplified to single variable functions of n:

$$\mathrm{Num}_{\mathrm{anchor}} = \sum_{k=0}^{k=n} (k+1) \times 6, \tag{15}$$

$$S_{\mathrm{unlocat}} = [18 + 12 \times (n-1)] \times \frac{a^2\sqrt{3}}{4}, \tag{16}$$

$$S_{\mathrm{locat}} = \begin{cases} 4 \times \dfrac{a^2\sqrt{3}}{4}, & n = 1 \\ 22 \times \dfrac{a^2\sqrt{3}}{4}, & n = 2 \\ \left[4 + 18 \times (n-1) + \displaystyle\sum_{k=1}^{k=n-2} 12 \times k\right] \times \dfrac{a^2\sqrt{3}}{4}, & n > 2\,, \end{cases} \tag{17}$$

$$S_{\mathrm{cover}} = \begin{cases} 22 \times \dfrac{a^2\sqrt{3}}{4}, & n = 1 \\ 22 \times \dfrac{a^2\sqrt{3}}{4}, & n = 2 \\ \left[22 + 18 \times (n-1) + \displaystyle\sum_{k=1}^{k=n-1} 12 \times k\right] \times \dfrac{a^2\sqrt{3}}{4}, & n > 1\,. \end{cases} \tag{18}$$

The utilization coefficient of equilateral triangle diagrams is the complement of ratio of S_{unlocat} to S_{cover}:

$$1 - \frac{S_{\mathrm{unlocat}}}{S_{\mathrm{cover}}} = \begin{cases} 1 - \dfrac{18}{22}, & n = 1 \\ 1 - \dfrac{18 + 12 \times (n-1)}{22 + 18 \times (n-1) + \sum_{k=1}^{k=n-1} 12 \times k}, & n > 1. \end{cases} \tag{19}$$

As (16) and (18) imply, S_{unlocat} grows linearly with n, and S_{cover} grows linearly with the power of n. Consequently, the increment of n will lead to the ratio of S_{unlocat} to S_{cover} close to zero. It means that the utilization coefficient of equilateral triangle diagrams will get close to one hundred percent as the scale of network increases. In Figure 5 the growth rate of S_{unlocat} is far less than the growth rate of S_{cover} as n increases. When the scale of cover area increases, the influence of unlocation area in the LETLA will rapidly vanish.

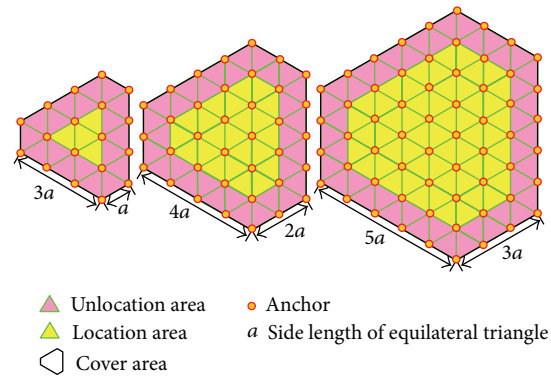


Figure 4: The topological structure of the equilateral triangle diagrams.

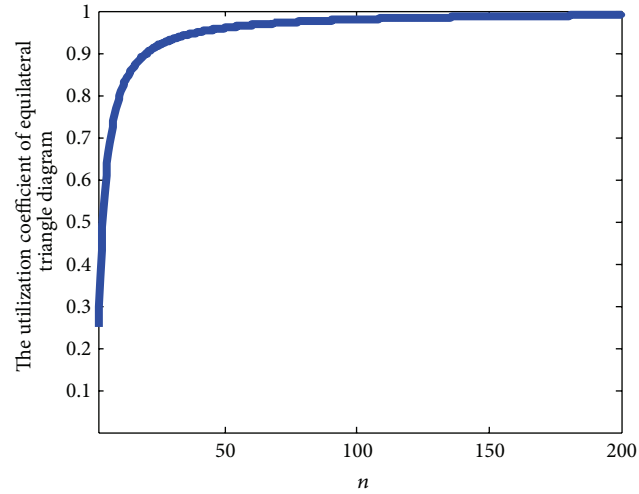


Figure 5: The utilization coefficient of equilateral triangle diagrams grows along with n's growth.

4. Lightness Equilateral Triangle Localization Algorithm

In this section, the LETLA will be explained by a practical example. This example is shown in Figure 1, and the given conditions are the same as Section 3. When the unknown node P knows the distance between itself and the anchor, the LETLA can be implemented as follows.

(a) Sort the ranging results and select the three nearest anchors A, B, and C, then find node P in $\triangle ABC$.

(b) $\triangle DEC$, $\triangle IBH$, and $\triangle AFG$ can be determined, for $\triangle ABC$ is their overlapped region and their side lengths are equal to $2a$. The geometric relation of equilateral triangles is determined by the deployment of anchors which is known by node P.

(c) Node P finds the regions in $\triangle DEC$, $\triangle IBH$, and $\triangle AFG$, respectively, through the relationship between the ranging results and r, as exemplified in Table 1.

(d) As node P is in the fifth region of $\triangle DEC$, the third region of $\triangle IBH$, and the sixth region of $\triangle AFG$ in Figure 1(c), the arc with the center F and the radius FP and the other arc with the center C and the radius CP are selected on the basis of Table 2.

(e) Derive two tangential equations of the arcs obtained in the previous step and solve the two tangential equations.

The main parameters of a tangential equation include the coordinates of the point of tangency and the slope of tangency. When the tangent is perpendicular to the line which goes through C and the point of tangency K, the slope of tangency can be derived by

$$\text{slope of tangent} = -\frac{x_c - x_k}{y_c - y_k}. \tag{20}$$

(x_c, y_c) and (x_k, y_k) are the coordinates of C and K, respectively. P knows (x_c, y_c), but does not know (x_k, y_k). In order to determine (x_k, y_k), a coordinate system is established in Figure 6.

The new coordinate system can be built as follows. First, move the original coordinate system to the positive direction of the x-axis with the distance x_c. Second, move the original coordinate system to the positive direction of the y-axis with the distance y_c. Finally, rotate the original coordinate system 180 degree clockwise. In the new coordinate system, the origin is C denoted by (x_c', y_c'), and the coordinates of K are (x_k', y_k'). The values of (x_c', y_c') and (x_k', y_k') are shown in (21), where CP is a known ranging result:

$$\begin{aligned} x_c' &= 0, \\ y_c' &= 0, \\ x_k' &= -CP, \\ y_k' &= 0. \end{aligned} \tag{21}$$

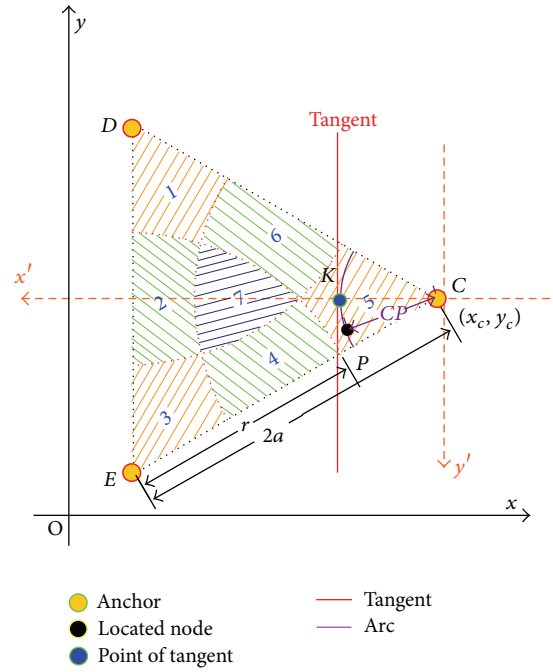


Figure 6: The calculation of the point of the tangent.

The relation of (x_k, y_k) and (x_k', y_k') is

$$\begin{aligned} x_k' &= -(x_k - x_c), \\ y_k' &= -(y_k - y_c). \end{aligned} \tag{22}$$

Because (x_c, y_c) and CP are known, combined with (21), (x_k, y_k) can be obtained by

$$\begin{aligned} x_k &= x_c - x_k', \\ y_k &= y_c - y_k'. \end{aligned} \tag{23}$$

The coordinates of the point of tangency are the function of the coordinates of the center of the arc, the length of radius of the arc, and the angle between the bisector and the x-axis. When the coordinates of the point of tangency are known, the slop of tangency can be drawn from (20). Now the tangential equation can be set up. Solving the intersection of two tangents is equivalent to solving the bivariate system of linear equations.

To sum up, the (a), (b), (c), and (d) in the LETLA are achieved by relational operation and only (e) is implemented by arithmetic operation. The procedure (e) only contains the four arithmetic operations and does not involve solving multivariable nonlinear equations, differentiation, integration, statistical estimation, and mathematical programming. It suggests that the LETLA is a lightness location method.

5. Performance Evaluation

5.1. Localization Error Evaluation. In Section 5.1, a numerical analysis of the performance of localization error will be given.

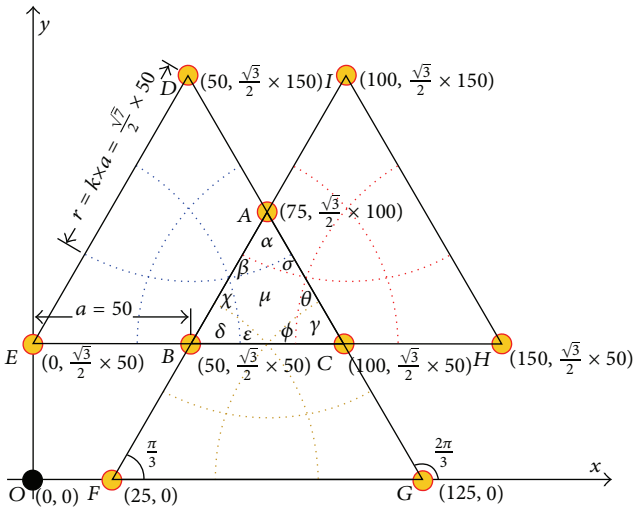


FIGURE 7: A numerical example of Figure 1(a).

Figure 7 shows the anchor distribution and the coordinate system. It is set up by putting Figure 1(a) in a coordinate system with an origin O on the lower left quarter. In Figure 7, the distance between a pair of adjacent anchors is denoted by a, which is the same in Figure 1(a) and assigned a value of 50. The length of radius to divide $\triangle DEC$, $\triangle IBH$, and $\triangle AFG$ is denoted by r, which is also like Figure 1(a) and assigned a value according to (2). To obtain the localization error $\text{Error}_{\text{locat}}$, node P has been placed in the different positions of $\triangle ABC$ for many times.

Figure 8 shows the distribution of $\text{Error}_{\text{locat}}$ in $\triangle ABC$. The z-axis in Figure 8 indicates the value of $\text{Error}_{\text{locat}}$ in the corresponding testing position. Given $a = 50$, the area of $\text{Error}_{\text{locat}} < 1$ makes up 72.971% of $\triangle ABC$. The area of $\text{Error}_{\text{locat}} < 3$ covers 90.519% of $\triangle ABC$. The maximum and minimum values of $\text{Error}_{\text{locat}}$ are 6.13 and 0, respectively. The data transmission radius of node P is l and $l > 2a$. A common evaluation method is computing the ratio of $\text{Error}_{\text{locat}}$ and l, and the ratio within 10% is a good indicator of localization performance [57]. Figure 8 illustrates LETLA can achieve the ratio within 10% in 92.655% of $\triangle ABC$. It means that the LETLA can obtain a good localization precision and a stable distribution.

In Figure 8, $\text{Error}_{\text{locat}}$ is associated with the division in $\triangle ABC$, denoted by α, β, σ, χ, δ, ε, γ, φ, θ, and μ as demonstrated in Figure 2(b) and defined in Table 3. The connecting points of β and χ, ε and φ, and σ and θ have the maximum value of $\text{Error}_{\text{locat}}$. $\text{Error}_{\text{locat}}$ in α, δ, and γ is all less than 1 and $\text{Error}_{\text{locat}}$ in μ is less than 1 in least 60% area. Moreover, $\text{Error}_{\text{locat}}$ is distributed uniformly in α, δ, γ, and μ. The reason of producing the maximum value of $\text{Error}_{\text{locat}}$ in the connecting point can be explained as follows. Combine Tables 2 and 3, when node P lies in the connecting point of the divisions in $\triangle ABC$, the intersection of two the arcs as determined from Table 2 is far away from the intersection of the two tangents. The reason is that the intersection of the two arcs is on the endpoint of the both arcs which both have a maximum of distance with the tangent. That result has been analyzed in Section 3.3 with the conception of $\text{Max}(\text{Error}_{\text{gap}})$ and the maximal value of $\text{Error}_{\text{gap}}$ is in the endpoint of the arc. Since the connecting point is the intersection of the two arcs' endpoint, $\text{Error}_{\text{gap}}$ of the two arcs reaches the maximum value simultaneously. Consequently, $\text{Error}_{\text{gap}}$ in the connecting point is greater than other positions of $\triangle ABC$ obviously.

Figure 9 shows the distribution of $\text{Error}_{\text{locat}}$ in $\triangle ABC$. Figures 8 and 9 are distinguished by the ranging noise. In Figure 8 $\text{Error}_{\text{locat}}$ is produced by the LETLA with accurate ranging results, but in Figure 9 $\text{Error}_{\text{locat}}$ is produced by the LETLA with the ranging results which are corrupted by noise following a Gaussian distribution. The ranging noise is represented by ζ, with expectation $E(\zeta) = 0$ and variance $D(\zeta) = 0.5$.

Given $a = 50$, $E(\zeta) = 0$, and $D(\zeta) = 0.5$, the area of $\text{Error}_{\text{locat}} < 1$ makes up 65.251% of $\triangle ABC$. The area of $\text{Error}_{\text{locat}} < 3$ covers 82.238% of $\triangle ABC$. The maximum and minimum values of $\text{Error}_{\text{locat}}$ are 7.26 and 0, respectively. Although the ranging noise lowers the localization precision of the LETLA, the LETLA also achieves the ratio within 10% in 90.14% of $\triangle ABC$.

Figure 9 shows that $\text{Error}_{\text{locat}}$ is associated with the division in $\triangle ABC$, which is the same as Figure 8. $\text{Error}_{\text{locat}}$ is distributed uniformly in α, δ, γ, and μ. In β, σ, χ, ε, φ, and θ, $\text{Error}_{\text{locat}}$ is affected by the noise, because the areas of these divisions are less than the areas of α, δ, γ, and μ. It leads to the misjudgment of the division including node P more easily:

$$P\{|\xi - E(\xi)| < 3D(\xi)\} \geq 0.8889. \tag{24}$$

By comparing Figures 9 with 8, the range of $\text{Error}_{\text{locat}}$ in Figure 8 and in Figure 9 is [0, 6.13] and [0, 7.26]. Based on the Chebyshev inequality of (24), $E(\zeta) = 0$, and $D(\zeta) = 0.5$, we obtain ζ in the range of $[-1.5, 1.5]$ with the probability of 88.89%. It means that the noise cannot change the range of $\text{Error}_{\text{locat}}$ obviously in Figure 8; hence the difference between the maximum of Figure 9 and that of Figure 8 is only 1.13. Consequently, the area of $\text{Error}_{\text{locat}} < 1$ and $\text{Error}_{\text{locat}} < 3$ only decreases 7.72% and 8.281%, respectively.

5.2. Localization Coverage and Beacon Density. To study the performances of the localization coverage and beacon density, we generate a variety of the test scenarios of the different numbers of anchors. The graphs in Figure 10 illustrate three test scenarios. The whole test area, designed as a circular domain and denoted by S_{test}, is the whole area of the test scenario. The whole test area is calculated using the following equation:

$$S_{\text{test}} = \pi \cdot (4a)^2. \tag{25}$$

The location area S_{locat} and the cover area S_{cover} represent the region in which the node can be located and region covered by anchors. For instance, Table 5 shows the number of anchors, the location area, and the cover area calculated for varying the side length of the cover area. The variable

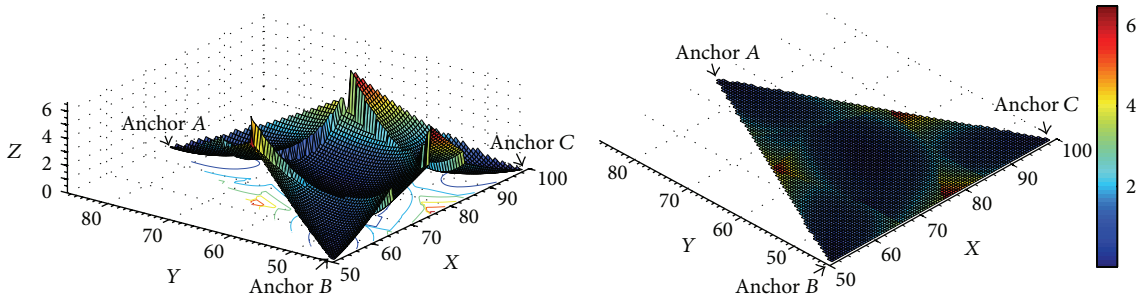


FIGURE 8: The distribution of $\text{Error}_{\text{locat}}$ in $\triangle ABC$ according to the geometrical relation in Figure 7: (a) the side view; (b) the top view. In both (a) and (b), the color indicates the value of $\text{Error}_{\text{locat}}$.

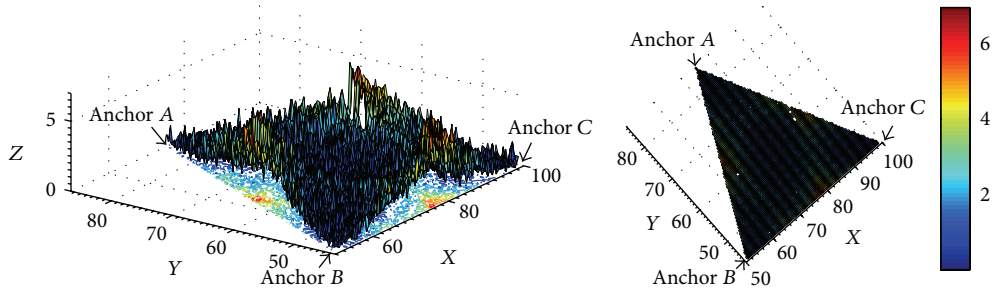


FIGURE 9: The distribution of $\text{Error}_{\text{locat}}$ in $\triangle ABC$ with ranging noise according to the geometrical relation in Figure 7: (a) the side view; (b) the top view.

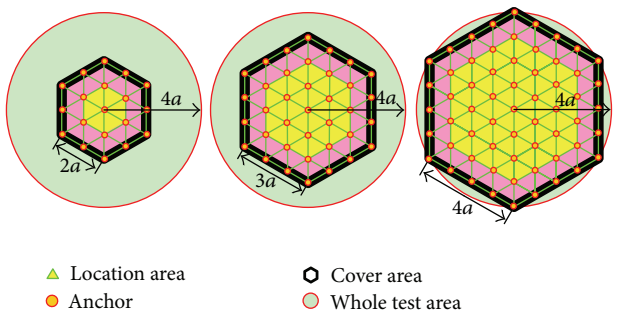


FIGURE 10: The topological structure of the test scenarios.

n is a factor used to indicate the side length of the cover area, as shown in Table 5. Because the number of anchors is determined by n, as shown in Table 5, we can replace the anchor number with the variable n to facilitate the calculation of the location area and cover area.

We use the following definitions in the evaluation: localization coverage, P_r, denotes the ratio of the location area to the whole test area:

$$P_r = \frac{S_{\text{locat}}}{S_{\text{test}}} = \frac{3\sqrt{3}\cdot(n-1)^2\cdot a^2}{2\pi\cdot(4a)^2} = \frac{3\sqrt{3}\cdot(n-1)^2}{32\pi},$$

$$P_d = \frac{\text{NUM}_{\text{anchor}}}{S_{\text{cover}}} = \frac{2\,(1+3n\cdot(n+1))}{3\sqrt{3}\cdot n^2\cdot a^2} = \frac{2+6n^2+6n}{3\sqrt{3}\cdot n^2\cdot a^2}. \quad (26)$$

Because the whole test area is a fixed circular domain, as expressed in (25) and illustrated in Figure 10, the localization coverage increases with growing anchor number, as shown in Figure 11(a). When the number of anchor increases to 91, the location area constitutes 82.699 percent of the test area. Figure 11(b) graphs the beacon density for varying the quantity of anchors under the condition of $a = 50$ which is the same as assigned in Section 5.1. We see that the beacon density falls sharply with increasing anchor number, until it reaches a density. We refer to this density as the saturation beacon density. The saturation beacon density can be obtained when the value of n approaches to infinity. If

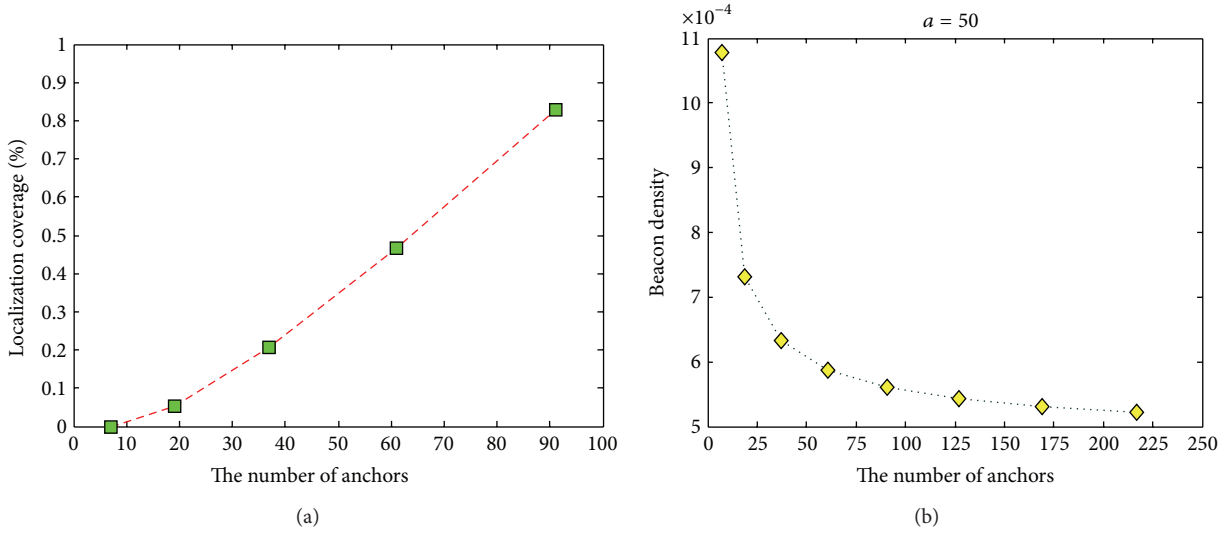


FIGURE 11: (a) The number of anchors versus localization coverage; (b) the number of anchors versus beacon density.

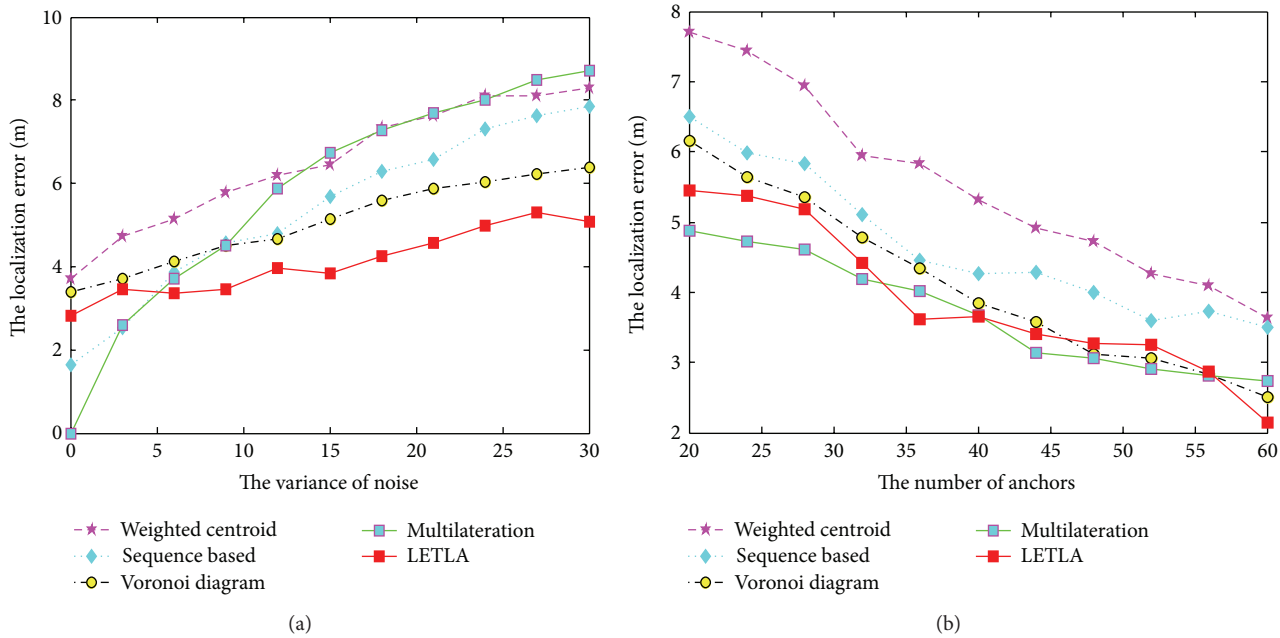


FIGURE 12: (a) The relationship between the localization error and the variance of noise; (b) the relationship between the localization error and the number of anchors.

TABLE 5: The relationship among the four indexes.

The side length of the cover area	The number of anchors	Location area	Cover area
$1a$	7	0	$3\sqrt{3}\cdot a^2/2$
$2a$	19	$3\sqrt{3}\cdot a^2/2$	$6\sqrt{3}\cdot a^2$
$3a$	37	$6\sqrt{3}\cdot a^2$	$27\sqrt{3}\cdot a^2/2$
$4a$	61	$27\sqrt{3}\cdot a^2/2$	$24\sqrt{3}\cdot a^2$
$5a$	91	$24\sqrt{3}\cdot a^2$	$125\sqrt{3}\cdot a^2/2$
$n\cdot a$	$1+3n\cdot(n+1)$	$3\sqrt{3}\cdot(n-1)^2\cdot a^2/2$	$3\sqrt{3}\cdot n^2\cdot a^2/2$

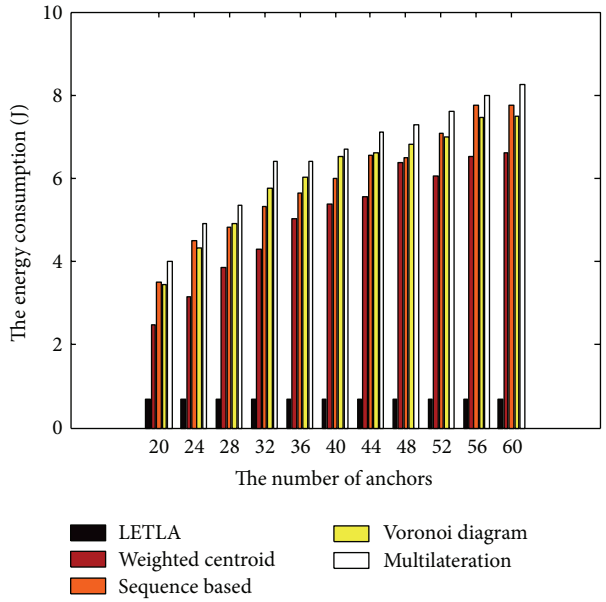


Figure 13: The energy consumption in the localization calculation versus the number of anchors.

Table 6: The simulation parameters of NS2.

Parameter	Value
Channel bandwidth (kbps)	250
Frequency (GHz)	2.4
Wireless model	Shadowing
MAC protocol	IEEE 802.15.4
Power for transmission (Watt)	0.28183815
Power for idle (Watt)	0.003587
Power for reception (Watt)	0.2182837
RX thresh (Watt)	$1.35685e-11$
RF radius (m)	50
Initial energy (J)	10

$a = 50$, the saturation beacon density will be $4.6188e-004$. We can reach the same conclusion in Figure 11(b):

$$\lim_{n\to\infty} P_d = \lim_{n\to\infty} \frac{2+6n^2+6n}{3\sqrt{3}\cdot n^2\cdot a^2} = \frac{2}{\sqrt{3}\cdot a^2}. \quad (27)$$

5.3. Comparisons. We compare the LETLA with the weighted centroid [33], the sequence based [36], the Voronoi diagrams [43], and the multilateration [58]. In order to study the performances of localization precision and energy consumption in different scenarios, NS2 is adopted in Table 6.

The anchors were deployed uniformly in a square field of 1000 m × 1000 m, and 1000 testing positions were selected to place the unknown node. The testing positions were evenly distributed across the 1000 m × 1000 m. The localization error was obtained by averaging the ten testing values of repeated measurements in each testing position. The anchors placements of the five location algorithms followed the geometric structure as Figure 4. To cover the square field of 1000 m × 1000 m, a portion of anchors were moved to the corners. In the experiments, we consider RSS as the range-measurement:

$$P_R(d) = P_T - PL(d_0) - 10\eta \log_{10}\frac{d}{d_0 - X_\sigma}. \quad (28)$$

The most widely used simulation model is the log-normal shadowing model [59], where P_R is the received signal power, P_T is the transmit power, and $P_L(d_0)$ is the path loss for a reference distance of d_0. η is the path loss exponent, and the random variation X_σ in RSS is generated according to Gaussian distribution with mean 0 and variance σ^2. In this model, we do not provision separately for any obstructions like walls. In order to compare the performance in energy consumption, we consider that a second in localization computing will consume $8.052518e-5J$. This value is obtained by dividing power for transmission into 3500. We modify the MAC layer of WPAN protocol in NS2 2.27 to realize the calculation of the energy consumption.

In Figure 12(a) the relationship between the localization error and the variance of noise is investigated when the number of anchors is 36. The result shows that the localization error increases with the variance of noise. We can observe that the performance of the multilateration is the best when the variance is less than six. As the variance increases, the LETLA is more robust than the other four algorithms under the same condition. Some of the reasons for this result can be briefly stated. The multilateration will obtain more precise coordinates if the variance becomes smaller. But its performance rapidly degrades when the effect of noise increases. As a result, the circles do not intersect at a common point and the precise coordinates degenerate into the approximate coordinates. The sequence based rises faster than the weighted centroid, the Voronoi diagrams, and the LETLA, since it needs a lot of iterative computations which are sensitive to the initial value. The weighted centroid and the Voronoi diagrams rise gently, but they are less accurate than the LETLA, because they have not fully utilized the information of the anchor placement.

In Figure 12(b), the localization error decreases with the increment of anchors, when the variance of noise is ten. As the number of anchors increases, the density of anchors also grows in the fixed cover area. For this reason the received signal power is enhanced with the decrement of the distances between the anchors and the unknown node. As shown in Figure 12(b), the sequence based, the LETLA, and the Voronoi diagrams descend in steps. The reason is that a few increments of the number of anchors are not sufficient to change the density obviously. Only some degrees of promotion can alter the density adequately and improve precision effectually. It is also interesting to note that the Multilateration is the best when the number of anchors less than 32. However, as anchors began to accumulate, the LETLA, the Voronoi diagrams, and the multilateration are of about the same accuracy.

Figure 13 presents the energy consumptions of five algorithms when the variance of noise is ten. Under the same condition, the multilateration consumes more energy than the other four methods. The multilateration with noisy data as

described above is based on solving nonlinear equations with complicated closed-form solutions. However, standard algorithms that provide least-square solutions for large numbers of nonlinear equations have a very high computational cost.

In Figure 13 the tendency of the weighted centroid, the sequence based, and the Voronoi diagrams rise with the number of anchors and the energy consumptions of these three methods are less than the multilateration but more than the LETLA. By studying the References [33, 36, 43], we can find some reasons. The Voronoi diagrams spend most of the computing energy in constructing Voronoi diagrams and carrying out iterative computations [43]. The sequence based consumes too many energy in the generalization of location sequence tables and the calculation of the Spearman's Rank Order Correlation Coefficient and the Kendall's Tau [36]. The weighted centroid does not contain iterative computation, so it is simpler than sequence based and Voronoi diagrams. However, the Weighted Centroid has more energy consumption for solving the constrained extremum problems [33].

On the other hand, the energy consumption of the LETLA is the least of all. As mentioned in Section 4, the LETLA only solves binary linear equations, except matching and sorting in a limited range, and does not contain any iterative computations, statistical equilibrium, and multivariate nonlinear equations. It is worthwhile to note that, the energy consumption of the LETLA is stable. There are two reasons. First, only two of the nine ranging results and the coordinates of anchors are used to construct and solve the binary linear equations. It means that the scale of the calculation of the LETLA is fixed and limited. Second, the matching and sorting operations can be implemented with the special data structure, like index point, which trades time for space. In contrast with the weighted Centroid, the sequence based, the Voronoi diagrams, and the multilateration, the results suggest that the LETLA is more energy efficient with the same localization accuracy.

6. Conclusion

In this paper, we presented a novel lightness equilateral triangle localization algorithm (LETLA). In the LETLA, the approximate coordinates were substituted for the real coordinates of the unknown node and an optimization problem was set up to minimize the approximate error. To solve this optimization problem, we proposed a new geometric structure called "equilateral triangle diagrams." In detail, we probed into the scalability of equilateral triangle diagrams and presented the geometrical characteristic of equilateral triangle diagrams. In the process of substituting the tangent for the arc, we derived that the midpoint of the arc was the most appropriate position of the point of the tangent.

Finally, the performance of the LETLA is much better than those of the other localization techniques, and experimental results indicate that the energy consumption of the LETLA is less than other location algorithm, at the same degree of location precision. As part of future work, we would focus on the implementation of the LETLA in the practical hardware of sensor nodes, such as MicaZ, Mica2, and Telos.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful comments which have significantly improved the quality of the paper. This work was supported in part by NSFC under Grant no. 60973022.

References

[1] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*, John Wiley & Sons, 2010.

[2] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 88–97, Atlanta, Ga, USA, September 2002.

[3] Q. Yang and Z. J. Wang, "Study on localization algorithms for large scale wireless sensor network," *Transducer and Microsystem Technologies*, vol. 26, no. 2, pp. 33–36, 2007.

[4] X. Wang, D. W. Bi, L. Ding, and S. Wang, "Agent collaborative target localization and classification in wireless sensor networks," *Sensors*, vol. 7, no. 8, pp. 1359–1386, 2007.

[5] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[6] W. Wang, V. Srinivasan, B. Wang, and K. C. Chua, "Coverage for target localization in wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 7, no. 2, pp. 667–676, 2008.

[7] Y. Qian, K. Lu, and D. Tipper, "A design for secure and survivable wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 5, pp. 30–37, 2007.

[8] J. You, Q. Han, D. Lieckfeldt, J. Salzmann, and D. Timmermann, "Virtual position based geographic routing for wireless sensor networks," *Computer Communications*, vol. 33, no. 11, pp. 1255–1265, 2010.

[9] M. F. Younis, K. Ghumman, and M. Eltoweissy, "Location-aware combinatorial key management scheme for clustered sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 8, pp. 865–882, 2006.

[10] A. Baggio and K. Langendoen, "Monte Carlo localization for mobile wireless sensor networks," *Ad Hoc Networks*, vol. 6, no. 5, pp. 718–733, 2008.

[11] A. Chehri, P. Fortier, and P. Martin Tardif, "UWB-based sensor networks for localization in mining environments," *Ad Hoc Networks*, vol. 7, no. 5, pp. 987–1000, 2009.

[12] L. Haibo, W. Yingna, and P. Bao, "A localization method of wireless sensor network based on two-hop focus," *Procedia Engineering*, vol. 15, pp. 2021–2025, 2011.

[13] J. Du and X. Zhang, "Research and improvement of localization algorithm for wireless sensor networks," *Energy Procedia*, vol. 13, pp. 8969–8975, 2011.

[14] M. Huang, S. Chen, and Y. Wang, "Minimum cost localization problem in wireless sensor networks," *Ad Hoc Networks*, vol. 9, no. 3, pp. 387–399, 2011.

[15] H. Aksu, D. Aksoy, and I. Korpeoglu, "A study of localization metrics: evaluation of position errors in wireless sensor networks," *Computer Networks*, vol. 55, no. 15, pp. 3562–3577, 2011.

[16] P. Shunmuga Perumal and V. Rhymend Uthariaraj, "Novel localization of sensor nodes in wireless sensor networks using

co-ordinate signal strength database," *Procedia Engineering*, vol. 30, pp. 662–668, 2012.

[17] B. Huang, C. Yu, and B. D. Anderson, "Analyzing localization errors in one-dimensional sensor networks," *Signal Processing*, vol. 92, no. 2, pp. 427–438, 2012.

[18] W. Hwa Liao, K. Ping Shih, and Y. Chee Lee, "A localization protocol with adaptive power control in wireless sensor networks," *Computer Communications*, vol. 31, no. 10, pp. 2496–2504, 2008.

[19] D. Ampeliotis and K. Berberidis, "Low complexity multiple acoustic source localization in sensor networks based on energy measurements," *Signal Processing*, vol. 90, no. 4, pp. 1300–1312, 2010.

[20] D. Niculescu and B. Nath, "DV based positioning in ad hoc networks," *Journal of Telecommunication Systems*, vol. 22, no. 1–4, pp. 267–280, 2003.

[21] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and 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)*, pp. 81–95, ACM, New York, NY, USA, September 2003.

[22] Y. Zhou, X. Ao, and S. X. Xia, "An improved APIT node self-localization algorithm in WSN," in *Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA '08)*, IEEE, Ed., pp. 7582–7586, IEEE, New York, NY, USA, June 2008.

[23] J. Z. Wang and H. G. Jin, "Improvement on APIT localization algorithms for wireless sensor networks," in *Proceedings of the International Conference on Networks Security, Wireless Communications and Trusted Computing (NSWCTC '09)*, Z. Hu and W. Li, Eds., pp. 719–723, IEEE Computer Society, Los Alamitos, CA, USA, April 2009.

[24] X. Q. Li, H. Gao, and L. L. Lv, "An improved APIT algorithm based on direction searching," in *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '09)*, IEEE, Ed., pp. 3268–3271, IEEE, New York, NY, USA, September 2009.

[25] X. F. Feng and H. B. Qi, "Improvement and simulation for a localization based on APIT," in *Proceedings of the International Conference on Computer Technology and Development (ICCTD '09)*, IEEE Computer Society, Ed., pp. 68–72, IEEE Computer Society, Los Alamitos, CA, USA, November 2009.

[26] W. Jia and J. Q. Fu, "Research on APIT and Monte Carlo method of localization algorithm for wireless sensor networks," in *Life System Modeling and Intelligent Computing*, K. Li, M. R. Fei, and L. Jia, Eds., pp. 128–137, Springer, Berlin, Germany, 2010.

[27] X. Y. Li, I. Stojmenovic, and Y. Wang, "Partial delaunay triangulation and degree limited localized bluetooth scatternet formation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 4, pp. 350–361, 2004.

[28] J. M. Font-Llagunes and J. A. Batlle, "Consistent triangulation for mobile robot localization using discontinuous angular measurements," *Robotics and Autonomous Systems*, vol. 57, no. 9, pp. 931–942, 2009.

[29] O. Tekdas and V. Isler, "Sensor placement for triangulation-based localization," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 3, pp. 681–685, 2010.

[30] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, 2000.

[31] J. Blumenthal, R. Grossmann, F. Golatowski, and D. Timmermann, "Weighted centroid localization in Zigbee-based sensor networks," in *Proceedings of IEEE International Symposium on Intelligent Signal Processing (WISP '07)*, J. U. Urena and J. J. G. Dominguez, Eds., pp. 1–6, IEEE, New York, NY, USA, October 2007.

[32] Q. Kong, X. Yang, and X. Dai, "Research of an improved weighted centroid localization algorithm and anchor distribution," in *Proceedings of the 2nd International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC '10)*, IEEE, Ed., pp. 400–405, IEEE, New York, NY, USA, October 2010.

[33] W. Jun, P. Urriza, H. Yuxing, and D. Cabric, "Weighted centroid localization algorithm: theoretical analysis and distributed implementation," *IEEE Transactions on Wireless Communications*, vol. 10, no. 10, pp. 3403–3413, 2011.

[34] K. Yedavalli, B. Krishnamachari, S. Ravulat, and B. Srinivasan, "Ecolocation: a sequence based technique for RF localization in wireless sensor networks," in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks (IPSN '05)*, IEEE, Ed., pp. 285–292, IEEE, New York, NY, USA, April 2005.

[35] Y. Yu, C. X. Jiang, X. J. Zhao et al., "Sequence Based localization algorithm with improved correlation metric and dynamic centroid," *Science China Information Sciences*, vol. 54, no. 11, pp. 2349–2358, 2011.

[36] K. Yedavalli and B. Krishnamachari, "Sequence-based localization in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 1, pp. 81–94, 2008.

[37] Z. Liu and J. Chen, "A new sequence-based iterative localization in wireless sensor networks," in *Proceedings of the International Conference on Information Engineering and Computer Science (ICIECS '09)*, pp. 1–4, December 2009.

[38] C.-C Hsiao and Y.-J. Tsai, "Node deployment strategy for WSN-based node-sequence localization," in *Proceedings of the 7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP '11)*, pp. 259–264, December 2011.

[39] D. Blanco, B. L. Boada, and L. Moreno, "Localization by Voronoi diagrams correlation," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '01)*, pp. 4232–4237, May 2001.

[40] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "A Voronoi approach for scalable and robust DV-hop localization system for sensor networks," in *Proceedings of the 16th International Conference on Computer Communications and Networks (ICCCN '07)*, pp. 497–502, August 2007.

[41] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. Loureiro, "DV-Loc: a scalable localization protocol using Voronoi diagrams for wireless sensor networks," *IEEE Wireless Communications*, vol. 16, no. 2, pp. 50–55, 2009.

[42] J. S. Li and H. C. Kao, "Distributed K-coverage self-location estimation scheme based on Voronoi diagram," *IET Communications*, vol. 4, no. 2, pp. 167–177, 2010.

[43] D. Ampeliotis and K. Berberidis, "Sorted order-K Voronoi diagrams for model-independent source localization in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 426–437, 2010.

[44] P. Biswas, T. C. Lian, T. C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Transactions on Sensor Networks*, vol. 2, no. 2, pp. 188–220, 2006.

[45] K. W. K. Lui, W. K. Ma, H. C. So, and F. K. W. Chan, "Semi-definite programming algorithms for sensor network

node localization with uncertainties in anchor positions and/or propagation speed," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 752–763, 2009.

[46] J. Lee, W. Chung, and E. Kim, "A new range-free localization method using quadratic programming," *Computer Communications*, vol. 34, no. 8, pp. 998–1010, 2011.

[47] W. Kim, J. G. Lee, and G. I. Jee, "The interior-point method for an optimal treatment of bias in trilateration location," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 4, pp. 1291–1301, 2006.

[48] Z. Mohamed, A. L. Mohamed, and B. Ridha, "Hybrid TOA/AOA approximate maximum likelihood mobile localization," *Journal of Electrical and Computer Engineering*, vol. 10, no. 1, pp. 1–5, 2010.

[49] Y. Cheng, X. Wang, T. Caelli et al., "Optimal nonlinear estimation for localization of wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 5674–5685, 2011.

[50] Z. Zhong and T. He, "Achieving range-free localization beyond connectivity," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*, pp. 281–294, IEEE, New York, NY, USA, November 2009.

[51] Z. Zhong and T. He, "Wireless sensor node localization by multisequence processing," *ACM Transactions on Embedded Computing Systems*, vol. 11, no. 1, pp. 1–33, 2012.

[52] J. Jeong, S. Guo, T. He, and D. H. C. Du, "Autonomous passive localization algorithm for road sensor networks," *IEEE Transactions on Computers*, vol. 60, no. 11, pp. 1622–1637, 2011.

[53] M. Li and Y. Liu, "Rendered path: range-free localization in anisotropic sensor networks with holes," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 320–332, 2010.

[54] L. Tang, X. Hong, and P. G. Bradford, "Privacy-preserving secure relative localization in vehicular networks," *Security and Communication Networks*, vol. 1, no. 3, pp. 195–204, 2008.

[55] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Secure localization algorithms for wireless sensor networks," *IEEE Communications Magazine*, vol. 46, no. 4, pp. 96–101, 2008.

[56] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[57] Y. Chen, W. Trappe, L. J. Greenstein, and S. Liu, "Non-interactive localization of cognitive radios based on dynamic signal strength mapping," in *Proceedings of the 6th International Conference on Wireless On-Demand Network Systems and Services (WONS '09)*, pp. 85–92, February 2009.

[58] M. Leonardi, A. Mathias, and G. Galati, "Closed form localization algorithms for mode S wide area multilateration," in *Proceedings of the 6th European Radar Conference (EuRAD '09)*, pp. 73–76, October 2009.

[59] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, 1999.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 620248, 10 pages
http://dx.doi.org/10.1155/2013/620248

Research Article

Range-Free Localization Scheme in Wireless Sensor Networks Based on Bilateration

Chi-Chang Chen, Chi-Yu Chang, and Yan-Nong Li

Department of Information Engineering, I-Shou University, Kaohsiung 84001, Taiwan

Correspondence should be addressed to Chi-Chang Chen; ccchen@isu.edu.tw

Received 4 October 2012; Accepted 17 December 2012

Academic Editor: Long Cheng

A low-cost yet effective localization scheme for wireless sensor networks (WSNs) is presented in this study. The proposed scheme uses only two anchor nodes and uses bilateration to estimate the coordinates of unknown nodes. Many localization algorithms for WSNs require the installation of extra components, such as a GPS, ultrasonic transceiver, and unidirectional antenna, to sensors. The proposed localization scheme is range-free (i.e., not demanding any extra devices for the sensors). In this scheme, two anchor nodes are installed at the bottom-left corner (Sink X) and the bottom-right corner (Sink Y) of a square monitored region of the WSN. Sensors are identified with the same minimum hop counts pair to Sink X and Sink Y to form a zone, and the estimated location of each unknown sensor is adjusted according to its relative position in the zone. This study compares the proposed scheme with the well-known DV-Hop method. Simulation results show that the proposed scheme outperforms the DV-Hop method in localization accuracy, communication cost, and computational complexity.

1. Introduction

Wireless sensor networks (WSNs) have gained worldwide attention in recent years. A WSN consists of spatially distributed autonomous sensors that cooperatively monitor a deployed region for physical or environmental conditions, such as temperature, sound, vibration, pressure, motion, and pollutants.

The manufacturing of small and energy efficient sensors has become technically and economically feasible because of recent advances in microelectromechanical systems (MEMSs) technology. A sensor node can sense, measure, and gather information from the environment and, based on some local decision process, transmit the sensed data to sinks (or base stations) through a wireless medium.

The transmission power consumed by a wireless radio is proportional to the distance squared or even a higher order in the presence of obstacles. Thus, multihop routing is usually considered for sending collected data to the sink instead of direct communication. Most WSN routing algorithms require the position information of sensor nodes. However, for some hazardous sensing environments, it is difficult to deploy the sensor nodes to the required locations. Thus, for environments in which it is difficult to plan the location of sensors in advance, localization techniques can be used to estimate sensor positions. The simplest and most common localization technique is to install a GPS receiver on each sensor in the sensor networks. Although the cost of GPS receivers is falling, they are still too costly, in price and energy consumption, to install in a sensor network.

This study proposes a low-cost yet effective WSN localization scheme. This scheme needs only two anchor nodes and uses low-complexity operations to estimate the location of unknown nodes. This study also compares the performance of the proposed scheme with the DV-Hop [1] method to show its superiority.

The rest of this paper is organized as follows. Section 2 reviews related research on WSN localization algorithms. Section 3 presents the communication protocol used to divide the deployed region into zones and the preliminary localization method. Section 4 introduces the more accurate enhanced method to estimate the positions of the unknown sensor nodes. Section 5 provides a simulation of the proposed localization scheme and a comparison of its performance

with the DV-Hop method. Finally, Section 6 offers a conclusion.

2. Related Work

Research interest in WSN localization has increased greatly in recent years. Traditional WSN localization technologies can be divided into two categories: *range-based methods* and *range-free methods* [2]. A range-based method positions the sensor nodes using additional devices, such as timers, signal strength receivers, directional antennas, and antenna arrays. In contrast, a range-free method requires no additional hardware and instead uses the properties of the wireless sensor network and the appropriate algorithms to obtain location information.

Range-based localization relies on the availability of point-to-point distance or angle information. The distance/angle information can be obtained by measuring time of arrival (ToA) [3], time difference of arrival (TDoA) [4], received signal strength indicator (RSSI) [4], and angle of arrival (AoA) [5]. Range-based localization may produce fine-grained resolution but places strict requirements on signal measurements and time synchronization.

Range-free localization requires no distance or angle measurements among nodes. This technique can be further divided into two categories: local techniques and hop-counting techniques [2].

In the local techniques, a node with unknown coordinates collects the position information of its neighbor beacon nodes with known coordinates to estimate its own coordinate. In the simple centroid algorithm proposed in [6], each sensor estimates its position as the centroid of the locations of the neighboring beacons. A density-adaptive algorithm can reduce the number of computation errors if beacons are well positioned [7]. However, this is unfeasible for ad hoc deployment. Later, He et al. proposed the APIT method [8], which divides the environment into triangular regions between beacon nodes. Each sensor determines its relative position based on the triangles and estimates its own location as the center of gravity of the intersection of all the triangles that the node may reside in. However, APIT requires long-range beacon stations and expensive high-power transmitters.

A hop-counting technique, called DV-Hop method, was proposed by Niculescu and Nath in [1]. In the DV-Hop method, each unknown node asks its neighboring beacon nodes to provide their estimated hop sizes and then attempts to obtain the smallest hop count to its neighbor beacon nodes using the designated routing protocol. Each unknown node estimates the distances to its neighbor beacon nodes by the hop counts to them and the hop size of the closest beacon node. The unknown nodes then apply trilateration to estimate their position based on the estimated distances to three suitable neighbor beacon nodes.

There are many followup studies of the DV-Hop method. The authors of [9] proposed the DV-Loc method, which shows how Voronoi diagrams can be used efficiently to scale a DV-Hop algorithm while maintaining or reducing the DV-Hop localization error. The main concept of the DV-Loc method is to use a Voronoi diagram to limit the scope of the flooding in a DV-Hop localization system. DV-Loc is a scalable solution that uses the Voronoi cell of a node to limit the region that is flooded when computing its position to reduce its localization error.

The authors of [10] proposed a range-free localization algorithm that uses expected hop progress to predict the location of WSN sensors. The algorithm is based on an analysis of hop progress in a WSN with randomly deployed

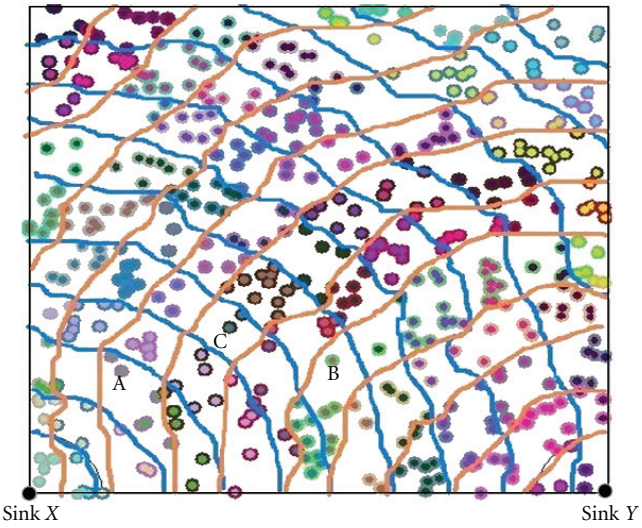


Figure 1: A scenario of 300 sensors with a communication range of 20 m, with the monitored region (200×200 m^2) divided into zones. The colored irregular arcs are added to facilitate visualization of the zones.

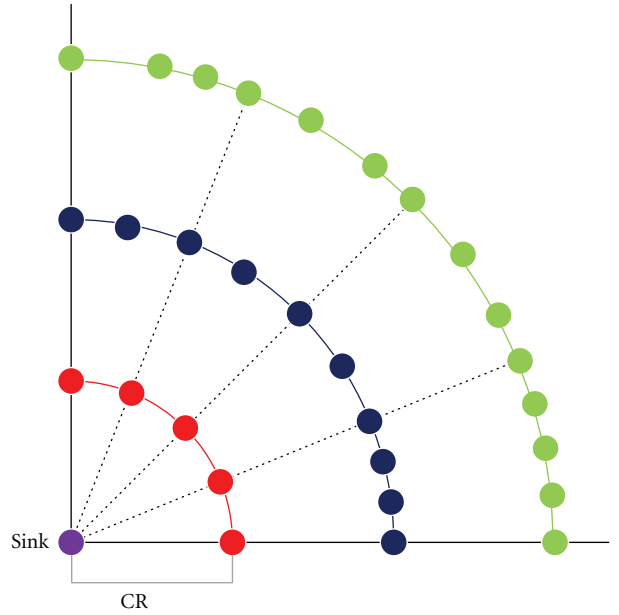


Figure 2: A scenario of maximum distance to the sink: sensor nodes are located at the edge of the communication range. Thus, the maximum distance of sensors from the sink with hop count n is $n \times$ CR, where CR is the communication range.

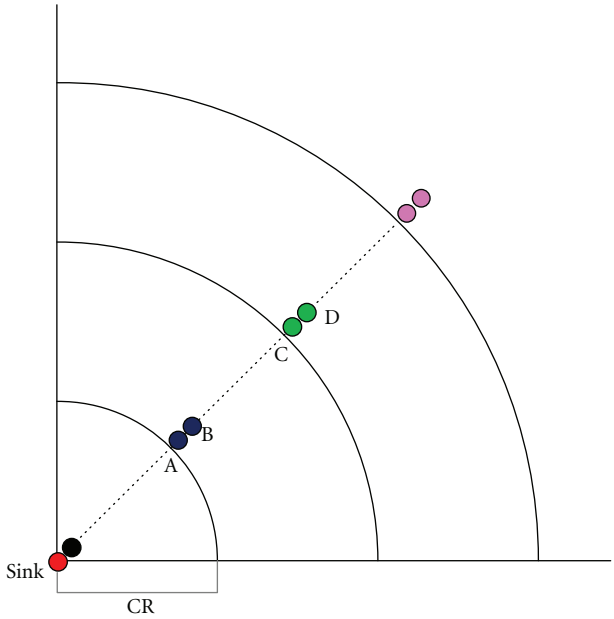


Figure 3: A scenario of minimum distance to the sink: sensor nodes are two in a group located close to the edge of the communication range.

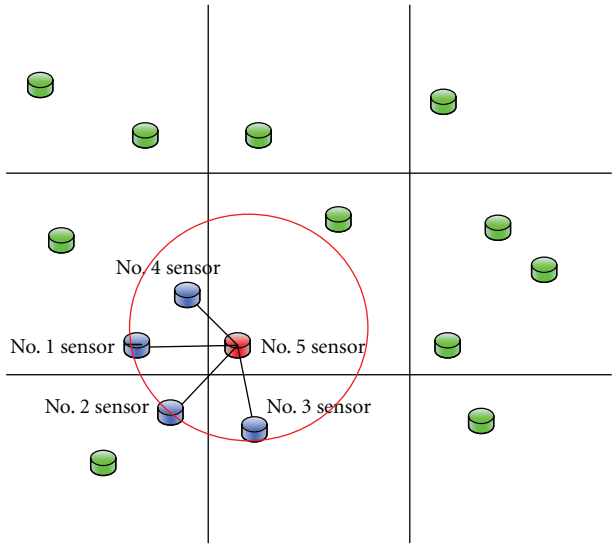


Figure 4: A scenario after broadcasting step. Sensor No. 5 is located at the southwest corner.

sensors and arbitrary node density. By deriving the expected hop progress from a network model for WSNs regarding network parameters, this system can compute the distance between any pair of sensors.

Traditionally, hop counts between any pair of nodes can only take an integer value, regardless of the relative positions of nodes in the hop. The authors of [11] argued that partitioning a node's one-hop neighbor set into three disjoint subsets according to their hop-count values can transform the integer hop count into a real number. The transformed real number hop count is then a more accurate representation of a node's relative position than an integer-valued hop count. In that paper, the author presented an algorithm termed HCQ (hop-count quantization) to perform this transformation.

Bilateration [12, 13], which is derived from multilateration, is based on the distance differences from an unknown node to two beacons at known locations. Unlike multilateration, which usually uses an iterative method to estimate the location of an unknown node, bilateration uses a basic geometry property, the intersection of two circles, to calculate the location of an unknown node. The computation of bilateration is much simpler than that of multilateration, which usually applies more expensive computation such as the least squares method. The disadvantage of bilateration is that the error rate is sensitive to the distance estimation to the beacon nodes.

In [12], Cota-Ruiz et al. presented a distributed and formula-based bilateration algorithm that can be used to provide an initial set of locations. In their scheme, each node uses distance estimates to beacons to solve a set of circle-circle intersection (CCI) problems, solved through a purely geometric formulation. The resulting CCIs are processed to pick those that cluster together, and the average is then used to estimate an initial node location. A similar bilateration algorithm was also proposed by the authors in [13] independently.

3. Zone-Based Localization Scheme

As mentioned in Section 1, some WSNs encounter difficulty in planning the location of sensors in advance. However, most routing algorithms require the information of sensor location. This section presents the proposed localization scheme to estimate the location of sensors. The following section extends the scheme to obtain a more accurate estimation of sensor locations.

The basic communication protocol used in the proposed scheme is flooding, which is a simple and effective mechanism for sending messages between sinks and sensor nodes. Flooding guarantees that sinks can reach any target node as long as the network is connective. In this scheme, the flooding mechanism serves as the initial routing step to acquire the minimum hop count to each sink for each sensor.

3.1. Localization Scheme. In the proposed localization scheme, called the zone-based localization method (ZBLM), two sink nodes are installed at the lower-left corner (Sink X) and the lower-right corner (Sink Y) of a square monitored region (Figure 1). This scheme assumes that (1) all the sensors are homogeneous, (2) they are uniformly deployed, and (3) the network is connective.

The ZBLM consists of three major steps: compute minimum hop counts, divide the monitored region into zones, and assign the coordinate of sensors for each zone.

Step 1 (Compute Minimum Hop Counts). First, we allow both Sink X and Sink Y to broadcast a hop-counting packet (HC packet in short) to their neighbor sensors. The HC packet contains two fields: (1) *Min_hc* (minimum hop count

to the source node), initialized to 0 and (2) the source node ID (0 for Sink X and 1 for Sink Y).

Each sensor records two current minimum hop count values (say, X_{hop} and Y_{hop}) to Sink X and Sink Y, respectively, which are both set to positive infinity initially. Once a sensor receives an HC packet, it checks the hop count field *Min_hc* in the HC packet. If *Min_hc* + 1 is smaller than its corresponding current minimum hop count value X_{hop} (or Y_{hop}), then it increases *Min_hc* by one before forwarding the packet to its neighbors and updates its X_{hop} (or Y_{hop}) to the new *Min_hc* value accordingly. Otherwise, the sensor discards the current incoming HC packet.

Step 2 (Divide the Monitored Region into Zones). After finishing the flooding of HC packets by Step 1, each sensor should have two minimum hop-count values (X_{hop}, Y_{hop}) for Sink X and Sink Y, respectively. Sensors with the same (X_{hop}, Y_{hop}) pair are in the same *zone* (note that the following subsection claims that the *zone* tends to be a geometry quadrilateral), and it is denoted as *zone* (X_{hop}, Y_{hop}). Figure 1 shows a scenario of dividing the monitored region into zones, in which the color irregular arcs are added for ease of visualization. In this figure, each node has its own (X_{hop}, Y_{hop}) pair. For example, X_{hop} of Node A is 3 and X_{hop} is 8. Therefore, Node A is in *zone* (3, 8). Similarly, Node B is in *zone* (6, 5), and Node C is in *zone* (5, 7).

Step 3 (Assign the Coordinate of Sensors for Each Zone). Although we have the hop counts of each sensor and, therefore, know which zone a sensor belongs to, this information is still insufficient for identifying the location of a given sensor. As shown in Figure 1, the distance of each hop is not necessarily the same, and thus the band width corresponding to a hop is not necessarily equal to the communication range. The next subsection analyzes the range of the distance to the sinks for a given sensor node with its minimum hop count values and gives the estimated distance to the sinks. The current subsection assumes that we already have the estimated distances to Sink X and Sink Y for each node.

Suppose that the coordinates of Sink X and Sink Y are $(0, 0)$ and $(w, 0)$, respectively, where w is the length of the square monitored region. Denote the distances from an unknown sensor S to Sink X and to Sink Y as d_x and d_y, respectively. The coordinate (x, y) of Sensor S is the intersection of two circles centered at $(0, 0)$ and $(w, 0)$, respectively. Therefore, (x, y) can be obtained using the following equations:

$$\begin{aligned} (x-0)^2 + (y-0)^2 &= d_x^2, \\ (x-w)^2 + (y-0)^2 &= d_y^2. \end{aligned} \tag{1}$$

Thus, $x = (d_x^2 - d_y^2 + w^2)/2w$ and $y = \pm\sqrt{d_x^2 - x^2}$.

Because sinks are installed at the lower left and lower right corners of the monitored region, we can only take the positive solution of y. Therefore, the coordinate of the unknown sensor S is

$$\left(\frac{d_x^2 - d_y^2 + w^2}{2w}, \sqrt{d_x^2 - \left(\frac{d_x^2 - d_y^2 + w^2}{2w} \right)^2} \right). \tag{2}$$

For (1) to produce a real solution, the sum of d_x and d_y (the radii of two circles) must be greater than or equal to w (the distance between two centers). This constraint is considered when estimating the distances d_x and d_y.

3.2. Estimate the Hop Distances between Sensors and the Sinks. At first, we claim that sensors with the same (X_{hop}, Y_{hop}) pair tend to congregate in a quadrilateral.

Suppose the length of the square monitored region is m, the communication range of each sensor is CR, and the total number of nodes is n. The probability that a node v is within the communication range of another given node u is $(\pi \times \text{CR}^2)/m^2$. Since the total number of nodes is n, the expected number of neighbor nodes, say ρ, for u is $(n-1)((\pi \times \text{CR}^2)/m^2)$. For example, if $m = 200$, CR = 30, and $n = 300$, then $\rho = (300 - 1) \times ((3.14 \times 30^2)/200^2) \cong 21$. Previous research [14] provided a more precise estimation of node degree that considers the border effect. According to [14], $\rho = (n-1)(\pi r^2 - (8/3)r^3 + ((11/3) - \pi)r^4)$, where $r = (\text{CR}/m)$. Therefore, $\rho \cong 19$ for this case.

According to [15], message forwarding between any two nodes through flooding occurs along the straight-line path with a probability greater than 0.85 if the number of neighbor nodes is greater than or equal to 15. According to the previous paragraph, if CR = 20, then ρ is greater than 15 as long as $n > 478$. Alternatively, if CR = 30, then ρ is greater than 15 as long as $n > 213$. Thus, if both the forwarding paths from Sink X and Sink Y progress along straight lines, then the sensors with the same (X_{hop}, Y_{hop}) pair tend to congregate and form a quadrilateral (called a "zone" in this paper). The experiment in this study shows that the "zone effect" still exists for the case CR = 20 and $n = 300$ ($\rho \cong 9$) (Figure 1).

The following discussion presents two extreme cases in which the message is forwarded along the straight line. The hop distance between sensors is related to the communication range and the density of the sensors in the region [14, 15]. For high density, each sensor has a certain number of sensors within its communication range. Therefore, for Sink X (or Sink Y), it is highly possible that sensors are located at the edge of the communication range. For the extreme case shown in Figure 2, sensor nodes always exist at the edge of the communication range of each hop from the sink. Therefore, assuming that the communication range is CR, the maximum distance of a sensor with hop count n from the sink is $n \times \text{CR}$.

The other extreme case occurs when the density of sensor nodes in the region is low and each sensor node has few neighbors, yet the network remains connective. As Figure 3 shows, sensor nodes are two in a group located close to the edge of the communication range. The first node in each group is within the communication range of the second node of the previous group, but immediately beyond the communication range of the first node of the previous group. Meanwhile, the second node in each group is immediately beyond the communication range of the second node of the previous group.

For example, in Figure 3, Node C is within the communication range of Node B but immediately beyond the communication range of Node A. Node D is immediately

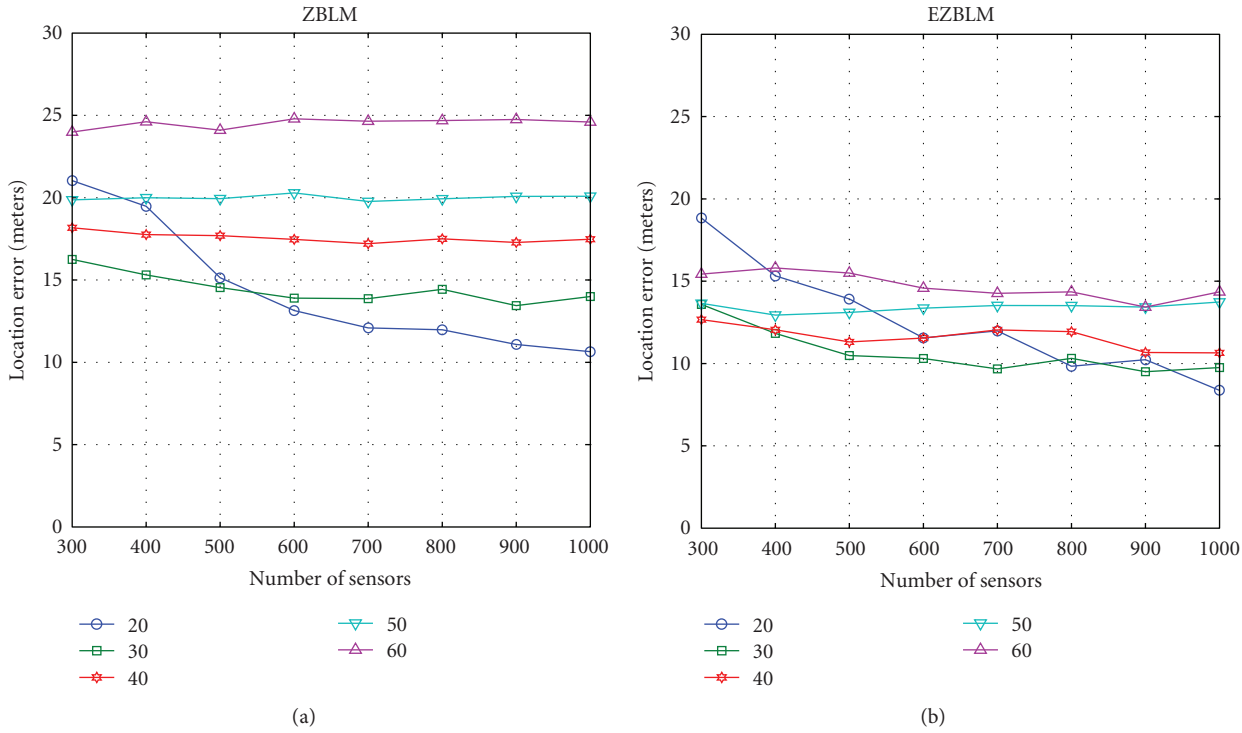


Figure 5: Location errors under different communication ranges (20–60 m) and node densities for ZBLM and EZBLM.

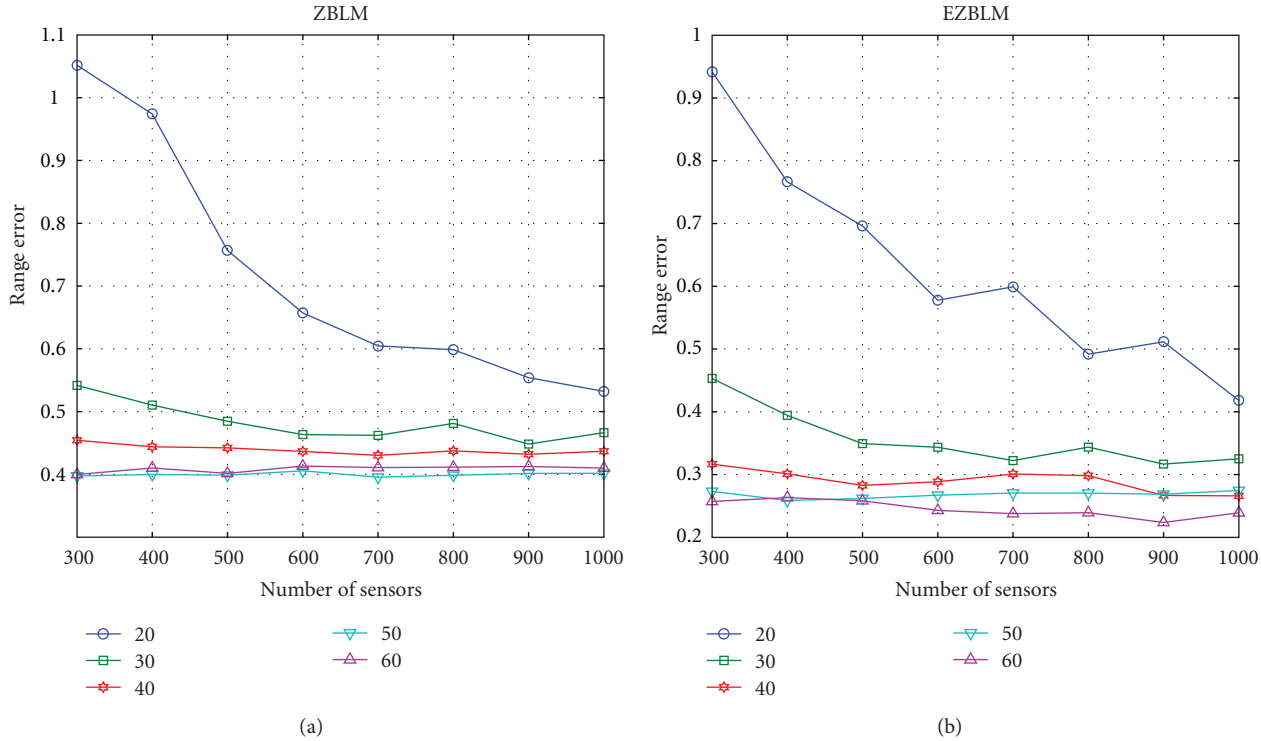


Figure 6: Range errors under different communication ranges (20–60 m) and node densities for ZBLM and EZBLM.

beyond the communication range of Node B. Thus, the minimum distance of sensors with hop count n is $\lfloor n/2 \rfloor \times \mathrm{CR} + \epsilon$, where ϵ is the distance between the two nodes in the same group. For example, the hop count of Node C is 4, the distance to the sink is $2 \times \mathrm{CR} + \epsilon$, the hop count of Node D is 5, and the distance is $2 \times \mathrm{CR} + \epsilon'$, where ϵ' is a value larger than ϵ. If the two nodes of each group are very close to each other yet still satisfy these conditions, then we can ignore the small value ϵ and say that the minimum distance of sensors with hop count n is $\lfloor n/2 \rfloor \times \mathrm{CR}$.

This analysis indicates that if the messages are forwarded along a straight line, the distance to the sink for any sensor with hop count n is between $\lfloor n/2 \rfloor \times \mathrm{CR}$ and $n \times \mathrm{CR}$ (ϵ is ignored). Therefore, if a sensor S is located in *zone* (m, n) (i.e., it has minimum hop count values (m, n) to Sink X and Sink Y), then we can use the following formula to approximate the distances, d_x and d_y, of sensor S *to* Sink X and Sink Y, respectively,

$$d_x = \left\{ \left\lfloor \frac{m}{2} \right\rfloor + \left[\left(m - \left\lfloor \frac{m}{2} \right\rfloor \right) * \alpha_1 \right] \right\} * \mathrm{CR},$$
$$d_y = \left\{ \left\lfloor \frac{n}{2} \right\rfloor + \left[\left(n - \left\lfloor \frac{n}{2} \right\rfloor \right) * \alpha_2 \right] \right\} * \mathrm{CR}, \quad (3)$$

where α_1 and α_2 are parameters between 0 and 1. For simplicity, this study uses the same value of α for both α_1 and α_2. To have a real solution for formula (1), it is necessary to rule out the α values that cause $d_x + d_y < w$. Section 5 shows that the value of α is related to the communication range and the density of the sensors and identifies the best α value that minimizes the location error of ZBLM for each condition in a WSN.

4. Enhanced Zone-Based Localization Method

The last section presents a localization scheme to estimate the positions of unknown sensors and prove that the sensors with the same hop-count pair tend to be clustered in the same zone. Sensors in the same zone have the same estimated coordinates. This can cause a certain amount of estimation error, unless these sensors are located at the same place, and the error increases as the area of each zone increases.

For convenience, this study calls the coordinate of a sensor obtained using the ZBLM scheme the ZBLM coordinate. This section proposes an adjustment algorithm, called the enhanced zone-based localization method (EZBLM), to reduce the estimation error. The basic concept of this algorithm is to determine the possible location of a sensor in the zone where it belongs and adjust the coordinate of the given sensor by the ZBLM coordinates of its relevant neighbor zones.

In a monitored region, each zone generally has eight neighbor zones, except for the boundary zones, which may have less neighbor zones (Figure 4). The following paragraphs detail how to determine which neighbor zones are closer to a given sensor in a zone, and how to adjust the coordinate.

Step 1. Each sensor uses half the communication range to broadcast a message that contains its ID, its hop-count pair to Sink X and Sink Y, and its ZBLM coordinate. (According to our simulation, the outcome of broadcasting using a half communication range is better than that of the full communication range, especially for sensors in the boundary zones.) Figure 4 shows a scenario after the broadcasting step. The blue sensors are within a half communication range of Sensor No. 5, indicating that Sensor No. 5 is close to its southwest neighbor zones.

Step 2. Each sensor that receives messages from neighbor nodes adjusts its coordinate according to the following steps.

(a) Extract the ZBLM coordinate from each received message, and consider the extracted coordinates (remove the duplicates) as a set of points S. Compute the centroid, say (x_c, y_c), of the points in S (i.e., take the arithmetic mean of all the points).

(b) Suppose the ZBLM coordinate of the sensor to be adjusted is (x_u, y_u). The adjusted coordinate is set to the center of the two coordinates (i.e., $(x_c + x_u)/2$, $(y_c + y_u)/2$).

The next section provides a comparison of the error rate of the coordinates obtained using both ZBLM and EZBLM, showing that EZBLM significantly improves the error rate of ZBLM.

5. Performance Analysis and Simulation Results

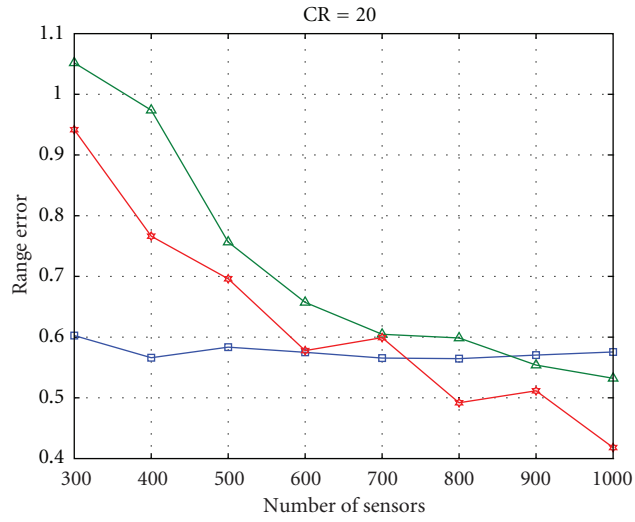
This section first identifies the values of α by experiments and suggests the best choice of the α value for each condition. We then compare two performance metrics, communication overhead, and computation overhead, for algorithms of the ZBLM, EZBLM, and DV-Hop. Finally, we simulate the three methods separately and compare their localization performance, including the location error and range error. The location error and range error are defined as follows.

$$\text{Location error} = \sqrt{(X_{\text{real}} - X_{\text{est}})^2 + (Y_{\text{real}} - Y_{\text{est}})^2},$$
$$\text{range error} = \frac{\text{location error}}{\text{CR}}, \quad (4)$$

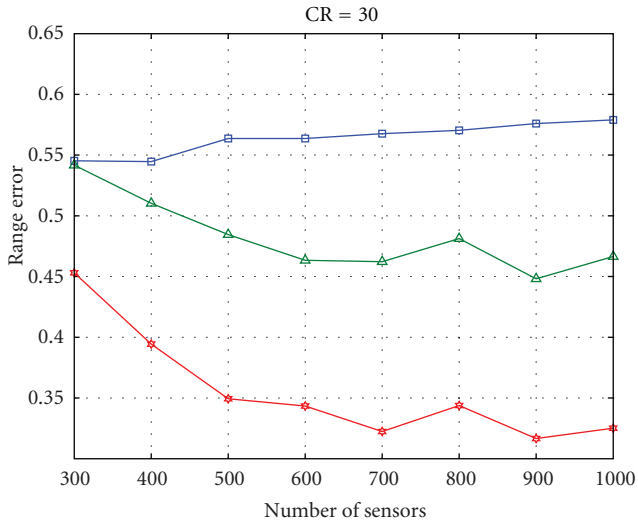
where $(X_{\text{real}}, Y_{\text{real}})$ and $(X_{\text{est}}, Y_{\text{est}})$ are real coordinates and estimated coordinates, respectively, of a given unknown sensor. CR is the communication range.

5.1. Determine the Value of α. The term α is a parameter used to estimate the hop distance for each sensor to the sinks using (3). The value of α represents the ratio of the estimated hop distance to the communication range and depends on the values of the communication range and the node density. However, as far as we know, no formula can calculate the exact value of α. Therefore, this study determines the value of α through experiments. The value of α is tested from 0.05 to 1.0 in 0.05 intervals. Each value of α is used to compute the location error of the ZBLM for each deployment. Table 1 lists the best α value that generates the least location error for each combination of communication range and node density over 500 different deployments.

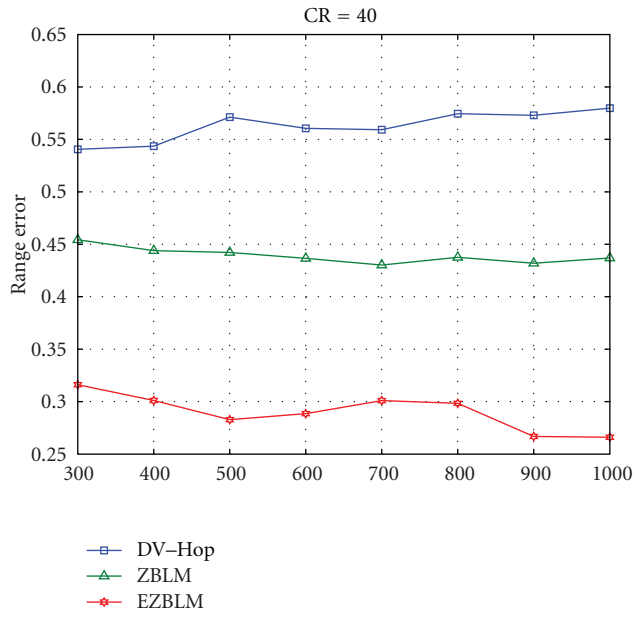
As Table 1 shows, most of the best α values lie between 0.6 and 0.75, except for the cases of low density (node number



(a)



(b)



(c)

Figure 7: Continued.

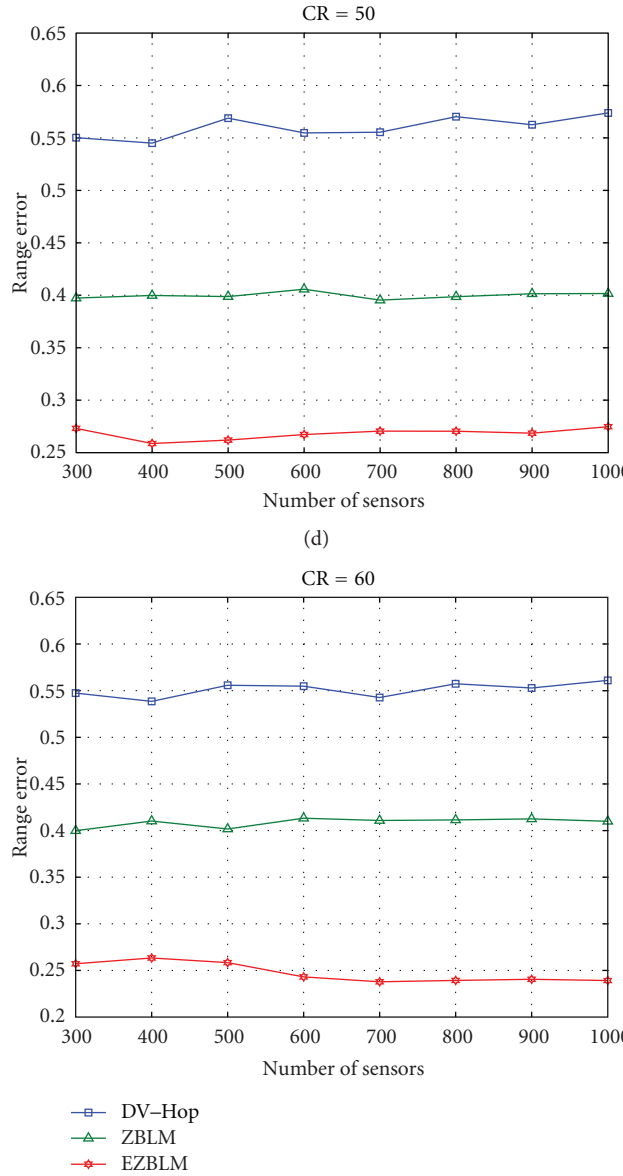


Figure 7: Range errors of the proposed methods (ZBLM and EZBLM) versus DV-Hop under different communication ranges.

Table 1: The best α value for different node densities and communication ranges.

Communication range (meters)		Number of sensors (node density)							
		300 (0.0075)	400 (0.01)	500 (0.0125)	500 (0.015)	600 (0.0175)	800 (0.02)	900 (0.0225)	1000 (0.025)
20	α Values of least location error	0.45	0.5	0.55	0.6	0.65	0.65	0.7	0.7
30		0.6	0.65	0.7	0.7	0.7	0.7	0.75	0.75
40		0.65	0.7	0.7	0.7	0.7	0.7	0.75	0.75
50		0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
60		0.65	0.65	0.65	0.7	0.7	0.7	0.7	0.7

Table 2: Performance comparison of the ZBLM, EZBLM, and DV-Hop (n is the total number of nodes).

Method	Communication cost	Computation cost (for each unknown node)	Number of anchor nodes
ZBLM	2 flooding operations	Constant	2
EZBLM	2 flooding operations + n broadcast operations	Constant	2
DV-Hop	$(1 + 20\%) \times n$ flooding operations	A variable number of iterations for trilateration	$20\% \times n$

≤500) and low communication range (CR = 20). The best α value increases in proportion to the node density under the same communication range. However, under the same node density, the best α value does not necessarily increase as the CR increases. This is because the proposed scheme uses an integral hop-count value, and the multiplication effect of the α value at a larger CR is more significant than that at a small CR. Therefore, larger α values for a larger CR may cause greater location error.

5.2. Performance Analysis. This section analyzes the performance of the proposed scheme in two respects. We first compare two performance metrics, communication overhead, and computation overhead, for algorithms of the ZBLM, EZBLM, and DV-Hop. We then simulate the three methods separately and compare their location errors and range errors.

According to the algorithm proposed in Section 3, the ZBLM individually needs two flooding operations from Sink X and Sink Y. The coordinate estimation simply computes the intersection of two circles, which takes constant time and uses basic arithmetic operations such as addition, multiplication, and the square root.

In addition to the flooding operations needed for the ZBLM, the EZBLM requires one broadcasting operation for each node to determine the position of each unknown node in its zone. The adjustment of coordinate uses two average operations, which takes constant time.

In the DV-Hop method (described in Section 2), each node (both beacon nodes and unknown nodes) needs one flooding operation to calculate the minimum hop counts to all other nodes and the hop size for each beacon node. Each beacon node needs one extra flooding to broadcast the hop size to all the unknown nodes. Therefore, this method requires (number of all nodes + number of beacon nodes) flooding operations. Each unknown node uses trilateration to estimate its location. The trilateration needs a variable number of iterations (from two to hundreds in our experiments) to converge to a point.

Both the ZBLM and EZBLM use only two anchor nodes. The simulations in [1] show that the DV-Hop method requires at least 20% of the sensors to be anchor nodes to obtain better results. Table 2 presents a performance comparison of these methods. The results show that the proposed methods outperform the DV-Hop method in communication cost, computational complexity, and the number of anchor nodes required.

5.3. Simulation Experiments. The simulation environments in this study were established as follows. The monitored region measured 200 m × 200 m. The number of sensors ranged from 300 to 1000, and the communication ranges are from 20 to 60 m. Sensors were uniformly deployed in the region. The α values were chosen from Table 1. Each simulation included 50 tests, with the mean serving as the final result. All the methods (ZBLM, EZBLM, and DV-Hop) were simulated using Matlab.

Figures 5 and 6 show the location errors and range errors of both the ZBLM and EZBLM, respectively, for different communication ranges and number of sensors. As expected, under the same communication range, the location error decreases as the sensor density increases for both the ZBLM and EZBLM. These simulation results show that the EZBLM improves the ZBLM scheme significantly.

For the simulation of the DV-Hop method, the rate of anchor nodes was set to 20% because its performance drops significantly when using less than 20% of anchor nodes [1]. Figure 7 shows that both the ZBLM and EZBLM outperform DV-Hop, except for the cases of CR = 20 and node number less than 500, in which each node may have too few neighbors and thus reduce the measurement accuracy of the proposed scheme. Note that the proposed scheme uses only two anchor nodes, whereas the DV-Hop method uses 20% of sensors as anchor nodes. These simulation results show that the proposed methods are more accurate than the well-known DV-Hop method.

6. Conclusions

Many studies have attempted to solve the range-free localization problems of WSNs. Most of them demand many anchor nodes and use the multilateration method, which requires complex computation and a variable number of iterations to estimate the location of sensors. This study proposes two range-free localization methods that use only two anchor nodes and apply the low-complexity bilateration method. Experimental results show that the range error of the EZBLM is less than 0.3 for all cases with a node density greater than 0.0075 (node number = 300 with 200∗200 region) when CR ≥ 40. Almost all the simulation results for the proposed method are better than those of the DV-Hop method, which requires many anchor nodes and more complex computations.

This study identifies the best α value to estimate the hop distance under different combinations of communication ranges and node densities. We show that sensors with the same minimum hop count pairs to Sink X and Sink Y tend to form a zone. Therefore, in addition to using the preliminary coordinate estimation method, the ZBLM, for unknown sensors, we use the EZBLM to adjust the coordinates of unknown sensors based on the sensor locations in the zones to which they belong. Simulation results show that this

adjustment significantly improves the location estimation performance for unknown sensors.

Although the proposed scheme uses a square monitoring region, the same algorithm can be applied to rectangular monitoring regions.

Acknowledgments

The authors are grateful for the support of I-Shou University under Grant ISU100-01-06 and the Ministry of Education under the Interdisciplinary Training Program for Talented College Students in Science, 100-B4-01.

References

[1] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *IEEE Global Telecommunicatins Conference (GLOBECOM'01)*, vol. 5, pp. 2926–2931, San Antonio, Tex, USA, November 2001.

[2] F. Liu, X. Cheng, D. Hua, and D. Chen, "TPSS: a time-based positioning scheme for sensor networks with short range beacons," in *Wireless Sensor Networks and Applications*, pp. 175–193, Springer, New York, NY, USA, 2008.

[3] H. Karl and A. Willig, *Protocols and Architecture for Wireless Sensor Network*, John Wiley & Sons, Hoboken, NJ, USA, 2005.

[4] A. Savvides, C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *Proceedings of the 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '01)*, pp. 166–179, Rome, Italy, July 2001.

[5] R. Peng and M. L. Sichitiu, "Angle of arrival localization for wireless sensor networks," in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad hoc Communications and Networks (SECON '06)*, pp. 374–382, Reston, Va, USA, September 2006.

[6] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, 2000.

[7] N. Bulusu, J. Heidemann, and D. Estrin, "Adaptive beacon placement," in *Proceedings of the 21st IEEE International Conference on Distributed Computing Systems (ICDCS-21' 01)*, pp. 489–498, Mesa, Ariz, USA, April 2001.

[8] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and 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)*, pp. 81–95, usa, September 2003.

[9] A. Boukerche, H. Oliveira, E. Nakamura, and A. Loureiro, "DV-Loc: a scalable localization protocol using Voronoi diagrams for wireless sensor networks," *IEEE Wireless Communications*, vol. 16, no. 2, pp. 50–55, 2009.

[10] Y. Wang, X. Wang, D. Wang, and D. P. Agrawal, "Range-free localization using expected hop progress in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 10, pp. 1540–1552, 2009.

[11] D. Ma, M. J. Er, B. Wang, and H. B. Lim, "Range-free wireless sensor networks localization based on hop-count quantization," *Telecommunication Systems*, vol. 50, no. 3, pp. 199–213, 2010.

[12] J. Cota-Ruiz, J.-G. Rosiles, E. Sifuentes, and P. Rivas-Perea, "A low-complexity geometric bilateration method for localization in wireless sensor networks and its comparison with least-squares methods," *Sensors*, vol. 12, no. 1, pp. 839–862, 2012.

[13] X. Li, B. Ha, Y. Shang, Y. Guo, and L. Yue, "Bilateration: an attack-resistant localization algorithm of wireless sensor network," in *Proceedings of the International Conference on Embedded and Ubiquitous Computing (IFIP '07)*, pp. 321–332, Taipei, Taiwan, 2007.

[14] K. Li, "Topological characteristics of random multihop wireless networks," in *Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops*, pp. 685–690, Providence, RI, USA, May 2003.

[15] J. Bachrach, R. Nagpal, M. Salib, and H. Shrobe, "Experimental results for and theoretical analysis of a self-organizing global coordinate system for ad hoc sensor networks," *Telecommunication Systems*, vol. 26, no. 2–4, pp. 213–233, 2004.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 758749, 8 pages
http://dx.doi.org/10.1155/2013/758749

Research Article

An Indoor Mobile Localization Strategy for Robot in NLOS Environment

Yan Wang, Yuanwei Jing, and Zixi Jia

The School of Information Science and Engineering, Northeastern University, Shenyang 110819, China

Correspondence should be addressed to Yan Wang; ywang8510@gmail.com

Received 4 October 2012; Accepted 17 December 2012

Academic Editor: Long Cheng

This paper deals with the problem of localization of mobile robot in indoor environment with mixed line-of-sight/nonline-of-sight (LOS/NLOS) conditions. To reduce the NLOS errors, a prior knowledge-based correction strategy (PKCS) is proposed to locate the robot. This strategy consists of two steps: NLOS identification and mitigation. We propose an NLOS identification method by applying the statistical theory. Then we correct the NLOS errors by subtracting the expected NLOS errors. Finally, the residual weighting algorithm is employed to estimate the location of the robot. Simulation results show that the proposed strategy significantly improves the accuracy of localization in mixed LOS/NLOS indoor environment.

1. Introduction

Indoor robot has been widely applied in health care, smart home, and emergency supporting. Location information is very important for mobile robot [1]. Due to the fact that GPS (Global Positioning System) does not work in indoor environment, so the indoor localization has motivated increased research interest. Since the Wireless Sensor Network (WSN) can be deployed quickly and flexible, so WSN is an effective application for indoor localization.

Recently, many indoor localization systems have been developed based on different wireless measurements: received signal strength (RSS) [2], time of arrival (TOA) [3], angle of arrival (AOA) [4], and time difference of arrival (TDOA) [5]. A mobile robot can estimate its location based on the wireless measurements with some beacon nodes whose locations are known prior. TDOA and AOA methods are energy-consuming resolution and they require extra hardware. TOA needs the high precision clock to achieve clock synchronization. So the above three methods are not suit for low configured sensor node. As an inexpensive approach, RSS has established the mathematical model on the basis of path loss attenuation with distance, and it requires relatively low configuration and energy. In this paper, we employ RSS localization method because of its characteristics. However, due to the complex indoor environment where walls and other obstacles are present, the wireless channel will always be blocked. The NLOS environment causes the radio signal to propagate a longer path than the true distance between the beacon nodes and target (e.g., robot or unknown node) due to the reflection and diffraction. So it induces the degradation of the localization accuracy. Thus, indoor localization which is robust to the NLOS environment is required [6]. In the presence of obstacles, the major positioning errors are from the measurements errors and NLOS propagation errors. And the NLOS effect leads to large localization errors if we do not consider it in the localization method.

For the purpose of realizing robust localization in NLOS environment, we propose a localization method using RSS measurement. And this method consists of four steps: NLOS identification, NLOS correction, Kalman filter, and residual weighting algorithm.

Several previous works have been investigated the localization algorithm in NLOS environment. The localization methods in NLOS environment can be categorized as nonparametric method and parametric method. For the nonparametric method, Chen [7] proposes a residual weighting algorithm to alleviate the NLOS errors in the location estimate. This method employs the least squares on all possible combinations of distance measurements and then calculates the final position as a weighted combination of

these intermediate estimates. The computational complexity of this method grows exponentially with the number of measurements. A robust multilateration algorithm [8] is introduced in NLOS environment. This algorithm is robust in comparison with traditional least squares multilateration. However, the performance of this method is severely affected by the number of NLOS measurements. Least squares support vector machine [9] classifier is employed to distinguish LOS/NLOS propagation and further mitigate the ranging errors in NLOS conditions. For the parametric method, the additive noise in the range measurement is characterized by different noise distributions corresponding to LOS and NLOS errors. In [10], Borras et al. propose a binary hypothesis test method to identify the NLOS measurements. In [11], Mazuelas et al. propose a prior NLOS measurements correction method to correct the measurements from NLOS propagation. Most of above works focus on the UWB system. But this system is not suit for WSN which consists hundreds of sensor nodes in the monitoring field.

Since the relatively low configuration and cost, RSS-based localization techniques have drawn considerable research interest [12]. An environment-adaptive method is investigated [13] that is tolerant to parameter variations caused by environmental variations. A sigma-point Kalman smoother-(SPKS) based location and tracking algorithm is proposed [14] for RSSI-based positioning and tracking. However, all of the above methods only consider the LOS environment. Fewer papers investigate the RSS-based localization method in NLOS environment.

In this paper, we firstly investigate the RSS-based NLOS identification method using the recorded measurements. Then the ratio of NLOS present in the record of measurements and the expectation of the NLOS errors are used to mitigate the NLOS errors. Kalman filter is employed to improve the estimated range. Finally, we use the residual weighting algorithm to estimate the location of the robot.

This paper is organized as follows. In Section 2, we introduce range estimation model. NLOS identification, NLOS mitigation, and localization method are presented in Section 3. Some simulation results present in Section 4. The conclusions are given in Section 5.

2. Range Estimation Model

The robot carries a sensor node which is used to communicate with the beacon nodes in the WSN. The beacon nodes emit signal continually. The robot estimates the distance between beacon node and the robot through received signal strength. The most widely applied signal propagation model is the log-normal shadowing model. The received signal strength of the robot from mth beacon node [15] is

$$\mathrm{PL}_m = \mathrm{PL}_0 - 10n_{\mathrm{los}/n\mathrm{los}} \log_{10}\left(\frac{d_m}{d_0}\right) + S_{\mathrm{los}/n\mathrm{los}}, \quad (1)$$

where PL_0 is the received signal strength at reference distance of d_0 meters. d_0 is assumed to be 1 m, and n is the path loss exponent. Generally line-of-sight of indoor environment shows n around 1.6 to 1.8, and around 4 to 6 in the presence of obstacles. $S_{\mathrm{los}} \sim N(0, \sigma_{\mathrm{los}}^2)$ is the LOS measurement noise modeled as zero mean white Gaussian with variance σ_{los}^2. $S_{n\mathrm{los}} \sim N(\mu_{n\mathrm{los}}, \sigma_{n\mathrm{los}}^2)$ is the measurement noise in NLOS environment. And $\sigma_{n\mathrm{los}} > \sigma_{\mathrm{los}}$. These parameters could be obtained in an environment through simulation or experiment. $d_m = \sqrt{(x_m - x)^2 + (y_m - y)^2}$ is the true distance between the mth beacon node and the robot. (x_m, y_m) is the coordinate of mth beacon node, and (x, y) is the position of the robot.

From (1), the estimated range between the mth beacon node and the robot can be expressed as

$$\begin{aligned}\widehat{d}_m &= 10^{(10n_{\mathrm{los}/n\mathrm{los}}\log_{10}d_m + S_{\mathrm{los}/n\mathrm{los}})/10n_{\mathrm{los}/n\mathrm{los}}} \\ &= d_m \cdot 10^{S_{\mathrm{los}/n\mathrm{los}}/10n_{\mathrm{los}/n\mathrm{los}}} \\ &= d_m + \left(10^{S_{\mathrm{los}/n\mathrm{los}}/10n_{\mathrm{los}/n\mathrm{los}}} - 1\right) d_m.\end{aligned} \quad (2)$$

We can approximately obtain that [16] $N_{\mathrm{los}} = (10^{S_{\mathrm{los}}/10n_{\mathrm{los}}} - 1)d_m \sim N(0, \sigma_1^2)$ under LOS environment, and $N_{n\mathrm{los}} = (10^{S_{n\mathrm{los}}/10n_{n\mathrm{los}}} - 1)d_m \sim N(\mu_2, \sigma_2^2)$ under NLOS environment.

So the estimated range can be rewritten as

$$\widehat{d}_m = d_m + N_{\mathrm{los}/n\mathrm{los}}. \quad (3)$$

Based on this range estimation model, we investigate NLOS identification when the obstacles present and propose a prior knowledge-based correction strategy (PKCS) to mitigate NLOS errors. The proposed method can be used to localize the mobile robot in indoor environment. This strategy will improve the localization accuracy.

3. NLOS Identification and Mitigation

The propagation conditions can change from LOS to NLOS when the robot is moving in the indoor environment. The NLOS propagation generally leads to a positive bias in the estimation range and causes a serious error in location estimation. So we should firstly identify the propagation conditions and then to mitigate the NLOS errors.

We make the following assumptions in the study. There are M beacon sensors and one robot in the field. The beacon nodes emit radio signal which attenuates inside the area under observation. The intensity of a signal emitted omni-directionally. The robot receives the radio and estimates the distance between the beacon node and the robot according to (3).

In this section, we propose a strategy for robot localization in NLOS environment. We firstly employ the polynomial fit to smooth the measurements, and then the average deviation between the smoothed curve and measurements and the standard deviation of the LOS noise are used to identify the NLOS measurements. Secondly, we estimate the NLOS ratio in the measurements to determine which measurements contain the NLOS errors, and then mitigate the NLOS measurements errors through subtracting the expectation of NLOS errors. Kalman filter is introduced to

improve the estimated range. Finally we use the residual weighting (Rwgh) algorithm to estimate the location of the robot. The flowchart of the proposed algorithm is given in Figure 1.

3.1. NLOS Identification. In order to identify the NLOS measurements, we consider the historical measurements from each beacon node individually. Based on the measurements of beacon node during a period of time and the standard deviation of the LOS noise, we determine whether the measurements contain the NLOS errors. We select a short period of time t_1–t_N, N is the number of sampled measurements from the mth beacon node, and the environment parameters are constant in the period. The estimated distance between the mth beacon node and the robot at time t_i is called $\widehat{d}_m(t_i)$. The standard deviation of the measurements of mth beacon node can be expressed as [17]

$$S = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\widehat{d}_m(t_i) - E\left(\widehat{d}_m(t_i)\right)\right]^2}. \tag{4}$$

We can determine that the data have affected by NLOS if the standard deviation is much larger than the standard deviation of the LOS noise. But the value of $E(\widehat{d}_m(t_i))$ is unknown, so we will employ a smooth version of the measurement records which we have stored to estimate the value of $E(\widehat{d}_m(t_i))$. For this purpose, we will use a smooth cubic regression of the measurements. The polynomial fit is used to estimate the deviation of the measurements. p_m is the third-order polynomial that best fits the available measurements minimizing its curvature. The third-order polynomial fulfills the objective of estimating the value of $E(\widehat{d}_m(t_i))$ with a high accuracy.

At mth beacon node, the measurements are smoothed as

$$p_m(t_i) = \sum_{k=1}^{3} b_m(k)\, t_i^k. \tag{5}$$

So the standard deviation of the measurements can be rewritten as

$$S = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\widehat{d}_m(t_i) - p_m(t_i)\right]^2}. \tag{6}$$

Because the NLOS errors are uncorrelated in time, so the measured data will have a significantly larger average deviation from the smoothed curve when the NLOS errors present. The algorithm requires comparison of the deviation S and the standard deviation of the LOS noise. If $S \gg \sigma_1$, then the measurements contain the NLOS errors. Otherwise the propagation path of the measurements is LOS.

3.2. NLOS Mitigation. When the NLOS is present, the measured data will deviate from the smoothed curve largely. In order to mitigate the NLOS errors, we must know which measurements are affected by NLOS. We firstly estimate the percentage of measurements in our records which are corrupted by NLOS errors during the sample period. Then we determine which measurements are corrupted by NLOS errors. Finally, an NLOS mitigation algorithm is proposed in this subsection.

If the measurements are affected by NLOS,

$$\begin{aligned}\left(\widehat{d}_{m,L}(t_i) - p_{m,L}(t_i)\right) &\sim N\left(\mu_L(K), \sigma_L^2(K)\right),\\ \left(\widehat{d}_{m,N}(t_i) - p_{m,N}(t_i)\right) &\sim N\left(\mu_N(K), \sigma_N^2(K)\right),\end{aligned} \tag{7}$$

where L and N represent the measurements taken at time t_i coming from LOS and NLOS, respectively. K is the percentage of measurements corrupted by NLOS errors. So

$$\begin{aligned} NS^2 &= \sum_{i=1}^{N}\left[\widehat{d}_m(t_i) - p_m(t_i)\right]^2\\ &= \sum_{\text{los}}\left[\widehat{d}_{m,L}(t_i) - p_{m,L}(t_i)\right]^2\\ &\quad + \sum_{n\text{los}}\left[\widehat{d}_{m,N}(t_i) - p_{m,N}(t_i)\right]^2\\ &= \sigma_L^2(K)\sum_{\text{los}}\left[\frac{\widehat{d}_{m,L}(t_i) - p_{m,L}(t_i)}{\sigma_L(K)}\right]^2\\ &\quad + \sigma_N^2(K)\sum_{n\text{los}}\left[\frac{\widehat{d}_{m,N}(t_i) - p_{m,N}(t_i)}{\sigma_N(K)}\right]^2\\ &= \sigma_L^2(K)\,X + \sigma_N^2(K)\,Y. \end{aligned} \tag{8}$$

Since

$$\begin{aligned}\frac{\widehat{d}_{m,L}(t_i) - p_{m,L}(t_i)}{\sigma_L(K)} &\sim N\left(\mu_L(K), 1\right),\\ \frac{\widehat{d}_{m,N}(t_i) - p_{m,N}(t_i)}{\sigma_N(K)} &\sim N\left(\mu_N(K), 1\right).\end{aligned} \tag{9}$$

From (9), it can be seen that the X and Y follow noncentral chi-square distribution: $X \sim \chi^{\text{NC}}_{(1-K)N,\lambda_1(K)}$, $Y \sim \chi^{\text{NC}}_{KN,\lambda_2(K)}$, where $\lambda_1 = \sqrt{\sum_{\text{los}}\mu_L^2(K)}$, $\lambda_2 = \sqrt{\sum_{n\text{los}}\mu_N^2(K)}$, NC means noncentral.

Therefore the expectation of the deviation can be presented as

$$\begin{aligned} E\left(S^2\right) &= \sigma_L^2(K)\left[(1-K) + \mu_L(K)\sqrt{\frac{1-K}{N}}\right]\\ &\quad + \sigma_N^2(K)\left[K + \mu_N(K)\sqrt{\frac{K}{N}}\right]. \end{aligned} \tag{10}$$

When the number of sample N is large, (10) can be rewritten as:

$$E\left(S^2\right) = (1-K)\,\sigma_L^2(K) + K\sigma_N^2(K). \tag{11}$$

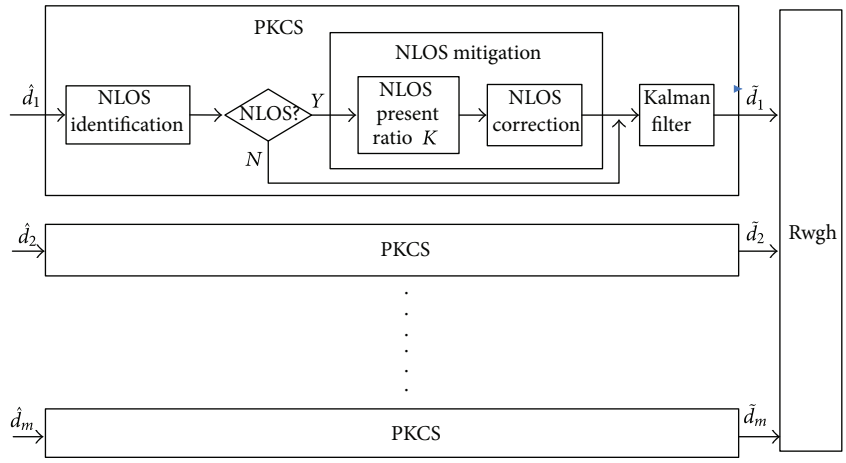


Figure 1: The architecture of the proposed algorithm frame.

The variance of S^2 is given by

$$\mathrm{Var}\left(S^2\right) = \frac{\sigma_L^4(K)\,\mathrm{Var}(X) + \sigma_N^4(K)\,\mathrm{Var}(Y)}{N^2}, \quad (12)$$

where $\mathrm{Var}(X) = 2(1-K)N + 4\mu_L(K)\sqrt{(1-K)N}$ and $\mathrm{Var}(Y) = 2KN + 4\mu_N(K)\sqrt{KN}$.

When N is relatively large, variance of S^2 is very small in (12). So we can assume that $S^2 \approx E(S^2)$. We can obtain S^2 as

$$S^2 \approx (1-K)\,\sigma_L^2(K) + K\sigma_N^2(K). \quad (13)$$

Assume that $\sigma_L(K) \approx \sigma_1$, $\sigma_N(K) \approx \sigma_2$. σ_2 is much larger than σ_1.

The percentage of measurements corrupted by NLOS errors is given by

$$K = \frac{S^2 - \sigma_1^2}{\sigma_2^2 - \sigma_1^2}. \quad (14)$$

Then we estimate which measurements are corrupted by NLOS errors after the percentage of measurements which have NLOS errors is obtained.

As we know, $P\{N_{\mathrm{los}} > (3/2)\sigma_1\} = 0.0668$. It means that the probability of LOS errors which larger than one and half its standard deviation is very low. So the large errors are mainly from the NLOS environment. And we assume that $p = P\{N_{n\mathrm{los}} > (3/2)\sigma_1\}$, so the number of NLOS measurements is pNK. We sort the standard deviation of the measurements S from large to small. The largest pNK measurements may contain the NLOS errors. We calculate the mean of these standard deviations of the pNK measurements, and it can be expressed as M_{NLOS}. Since the NLOS errors are the positive bias. We correct the pNK measurements which contain the NLOS by substracting the expected NLOS errors M_{NLOS}. The corrected range can be expressed as

$$\hat{d}_m'(q) = \hat{d}_m(q) - M_{\mathrm{NLOS}}, \quad q = 1, \ldots, \mathrm{pNK}. \quad (15)$$

3.3. Kalman Filter. The KF is one of the popular filtering methods developed and it, as well as its descendants, has been the dominate filter type for the past 40 years. Most localization systems use Kalman filter for state estimation. Kalman filter produces optimal estimates when measurement noise is Gaussian and stationary. It can be executed within a computer and iteratively without any extensive modification or special hardware. We perform the Kalman filter for the measurements in LOS and the corrected measurements in NLOS. The state equation of mth beacon node under LOS/NLOS environment can be expressed as follows:

$$X_m(t_{i+1}) = F(\Delta t)\,X_m(t_i) + Cw_m(t_i), \quad m = 1, \ldots, M, \quad (16)$$

where $X_m(t_i)$ and $X_m(t_{i+1})$ are the state vectors at time t_i and t_{i+1}, respectively. Let the state vector be $X_m(t_i) = [d_m(t_i)\ \ \dot{d}_m(t_i)]^T$, $m = 1, \ldots, M$, where T denotes the transpose operator, and $d_m(t_i)$ represents the distance between the mth beacon node and the robot. $\dot{d}_m(t_i)$ denotes the velocity of the robot, where $w_m(t_i)$ is the process noise and follows zero mean independently and identically distributed Gaussian with variance σ_m^2

$$F(\Delta t) = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \qquad C = \begin{bmatrix} \dfrac{\Delta t^2}{2} \\ \Delta t \end{bmatrix}, \quad (17)$$

where $\Delta t = 1$ is the sample period.

The measurement equation of sensor m under LOS/NLOS environment can be expressed as follows:

$$Z_m(t_{i+1}) = \hat{d}_m(t_{i+1}) = d_m(t_{i+1}) + N_{\mathrm{los}}, \quad m = 1, \ldots, M, \quad (18)$$

where $\hat{d}_m$ is the corrected range according to (15) in NLOS or the measurements in LOS.

We can obtain the predicted state and prediction covariance as follows:

$$\begin{aligned} \widehat{X}_m(t_{i+1} \mid t_i) &= F(\Delta t)\,\widehat{X}_m(t_i \mid t_i), \\ P_m(t_{i+1} \mid t_i) &= F(\Delta t)\,P_m(t_i \mid t_i)\,F^T(\Delta t) + \sigma_m^2 CC^T, \\ \gamma_m(t_{i+1}) &= Z_m(t_{i+1}) - \widehat{Z}_m(t_{i+1} \mid t_i), \\ \widehat{Z}_m(t_{i+1} \mid t_i) &= G\widehat{X}_m(t_{i+1} \mid t_i), \end{aligned} \tag{19}$$

where $\widehat{X}_m$ represents the estimate of state vector X_m, $\widehat{X}_m(t_{i+1} \mid t_i)$ is the predicted state estimate of state vector, $\widehat{X}_m(t_i \mid t_i)$ is the state estimate at the time t_i, $P_m(t_i \mid t_i)$ and $P_m(t_{i+1} \mid t_i)$ are the estimate and predicted covariance at the time t_i, respectively, and $\widehat{Z}_m(t_{i+1} \mid t_i)$ is the priori state estimate of measurement $Z(t_{i+1})$.

$G = [1\ \ 0]$, where G assigns different value according to different measurement mechanisms. Consider

$$S_m(t_{i+1}) = GP_m(t_{i+1} \mid t_i)\,G^T + Q_m, \tag{20}$$

where Q_m is the covariance matrix of measurement error. The Kalman gain is as follows:

$$K_m(t_{i+1}) = P_m(t_{i+1} \mid t_i)\,G^T S_m^{-1}(t_{i+1}). \tag{21}$$

Then we can obtain the updated state estimate and updated estimate covariance as follows:

$$\begin{aligned} \widehat{X}_m(t_{i+1} \mid t_{i+1}) &= \widehat{X}_m(t_{i+1} \mid t_i) + K_m(t_{i+1})\,\gamma_m(t_{i+1}), \\ P_m(t_{i+1} \mid t_{i+1}) &= P_m(t_{i+1} \mid t_i) - K_m(t_{i+1})\,GP_m(t_{i+1} \mid t_i). \end{aligned} \tag{22}$$

The ranging estimation in Kalman filter is expressed as:

$$\widetilde{d}_m(t_{i+1}) = D\widehat{X}_m(t_{i+1} \mid t_{i+1}), \quad D = [1, 0]. \tag{23}$$

3.4. Location Estimation. The outputs of Kalman filter are the estimated distances between the robot and beacon nodes. According to the estimated distances, we can locate the robot in the field. In this section, we introduce residual weighted localization method to estimate the location of the robot.

The Conventional Residual Weighting algorithm can be described as follows [7].

(a) Make initialization with $n = M$, $(M > 3)$. M is the number of sensor nodes, the beacon nodes form $Z = \sum_{i=3}^{M} C_M^i$ range measurement combinations. Each combination is represented by a beacon node index set $\{S_k \mid k = 1, 2, \ldots, Z\}$.

(b) For each beacon node index set, we employ the maximum likelihood method to estimate the location of the robot which can be represented as $\widehat{X}_k = [\widehat{x}_k, \widehat{y}_k]$. Then we compute the average of Res (residual) as follows:

$$\overline{\mathrm{Res}}\left(\widehat{X}_k, S_k\right) = \frac{\mathrm{Res}\left(\widehat{X}_k, S_k\right)}{\mathrm{size}\left(S_k\right)}, \tag{24}$$

where $\mathrm{Res}(\widehat{X}_k, S_k) = \sum_{j \in S_k} \left(\widehat{d}_j' - \sqrt{(x_j - \widehat{x}_k)^2 + (y_j - \widehat{y}_k)^2}\right)^2$ and (x_j, y_j) are the coordinate of jth beacon node.

(c) Find the final estimated location of the robot as the weighted linear combination of the intermediate estimations from step (b). The weight is inversely proportional to $\overline{\mathrm{Res}}$ of the estimation. Mathematically, we can get the estimated location of the robot as follows:

$$\widehat{X} = \frac{\sum_{k=1}^{Z} \widehat{X}_k \left(\overline{\mathrm{Res}}\left(\widehat{X}_k, S_k\right)\right)^{-1}}{\sum_{k=1}^{Z} \left(\overline{\mathrm{Res}}\left(\widehat{X}_k, S_k\right)\right)^{-1}}. \tag{25}$$

Table 1: Default parameter values.

Parameters	Default values
Number of sensor nodes (M)	6
Mean of the LOS noise (μ_L)	0
Mean of the NLOS noise (μ_N)	3
Standard variance of LOS noise (σ_L)	1
Standard variance of NLOS noise (σ_N)	5

4. Performance Evaluation and Analysis

In this section, we evaluate the performance of the NLOS localization method described in the previous sections through simulations. Table 1 presents the default parameter values in the experiments. We consider a 200 m × 160 m area with six beacon nodes, and the robot moves in the field. There are two obstacles in the field. In each simulation case, 2000 Monte Carlo runs are performed with the same parameters. The performance of the proposed algorithm is measured by average localization error as

$$\text{error} = \frac{1}{N_t} \sum_{i=1}^{N_t} \sqrt{(\widehat{x}_i - x)^2 + (\widehat{y}_i - y)^2}, \tag{26}$$

where $N_t = 2000$, (x, y) is the true location of the robot. $(\widehat{x}_i, \widehat{y}_i)$ is the estimated location of the robot at ith Monte Carlo trial.

In this section, we compare the proposed algorithms (PKCS-ML and PKCS-Rwgh) with the naïve ML and Rwgh algorithms. All of the four algorithms employ the Kalman filter to improve the estimated range.

In Figure 2, we compare the localization results between the Rwgh and PKCS-Rwgh algorithm. Figure 2(a) shows the sight state with respect to all the beacon nodes in sample points. We can see that the sight states vary with time. A mobile localization is shown in Figure 2(b). This figure compares the true trajectory between the estimated one obtained by Rwgh and another obtained by our method. The localization errors can be appreciated by looking at the lines that connect the true and the estimated positions. In Figure 2(c), we can obtain the detailed localization errors in each sample point. It can be observed that the proposed method has better performance in comparison with Rwgh in most of the sample points.

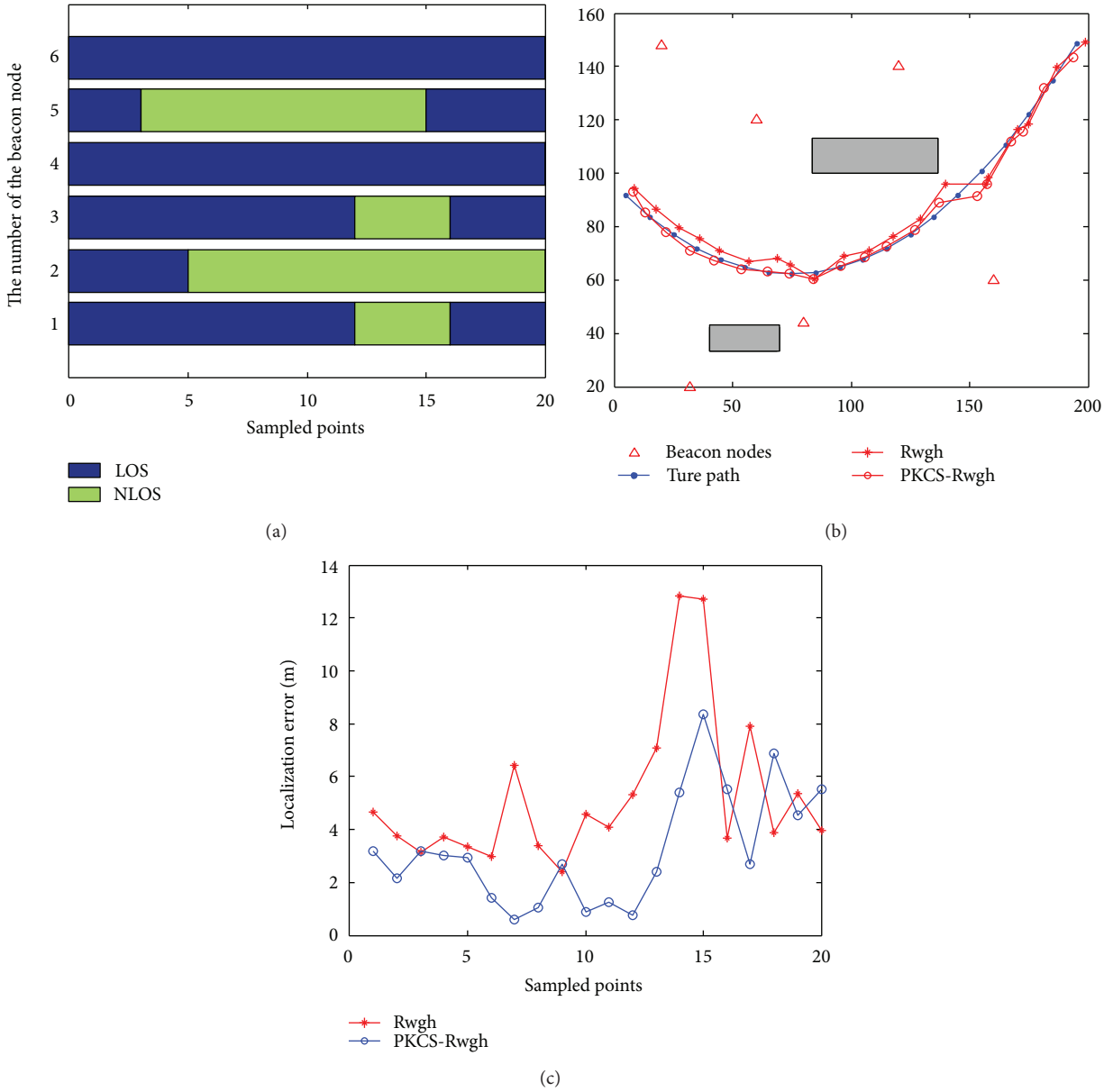


FIGURE 2: (a) The sight state in each sample point, (b) example of the mobile localization, (c) the localization errors in each sample point.

In Figure 3, the number of beacon nodes is varied from 4 to 8. The MLE and Rwgh methods with the prior knowledge-based correction strategy are called PKCS-MLE and PKCS-Rwgh, respectively. The results show that the effect of increasing the number of beacon nodes is higher using MLE method in comparison with the Rwgh method. However, the PKCS-MLE and PKCS-Rwgh methods have higher localization accuracy compared with MLE and Rwgh methods.

Figure 4 shows the average localization error versus the standard variance of LOS noise. The PKCS-MLE and PKCS-Rwgh methods demonstrate similar robustness against the standard variance of LOS noise. The localization errors of the MLE are always larger than 4 m and increase significantly as the standard variance of LOS noise increases. Due to the PKCS, the performance of PKCS-MLE and PKCS-Rwgh methods outperforms the MLE and Rwgh methods.

To investigate the robustness of the proposed method, we evaluate it under different standard variances of NLOS noise. In Figure 5, it is obvious that the average localization error increases with the increment of standard variance of NLOS noise. We can see that the localization accuracy of PKCS-MLE is improved significantly in comparison with MLE. And the PKCS-Rwgh has 6.87% higher localization accuracy than Rwgh.

In Figure 6, we show the average localization error versus mean of NLOS noise. The accuracy of all methods is decreased with the increase of mean of the NLOS noise. We can see that the proposed method can significantly

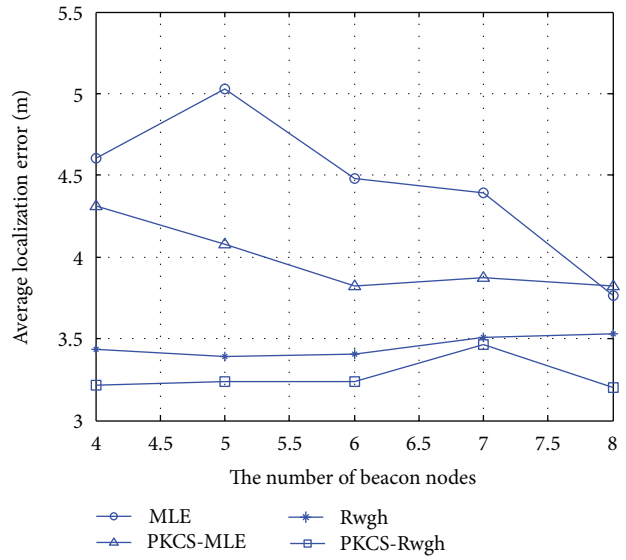


Figure 3: Average localization error versus the number of beacon nodes.

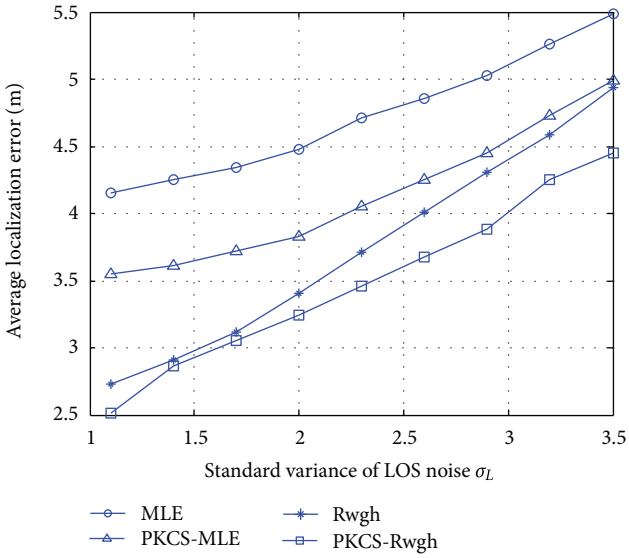


Figure 4: Average localization error versus the standard variance of LOS noise.

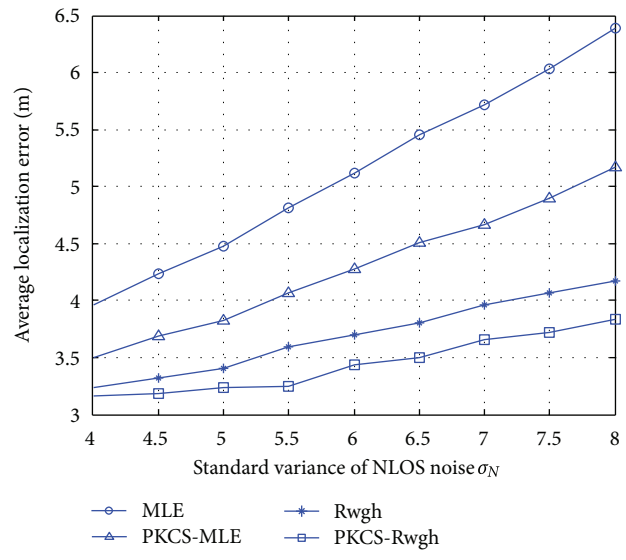


Figure 5: Average localization error versus the standard variance of NLOS noise.

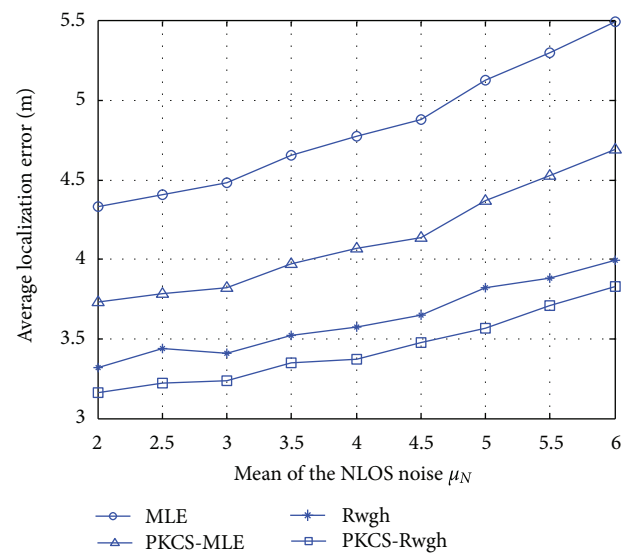


Figure 6: Average localization error versus the mean of NLOS noise.

improve the location estimation accuracy by subtracting the expectation of NLOS errors. And the average localization error of MLE and Rwgh methods are worse without prior knowledge-based correction strategy.

5. Conclusion

An NLOS correction approach has been proposed to locate the position of mobile robot in order to alleviate the NLOS errors which arise in clustering environment. We consider the prior knowledge of the errors to identify the NLOS and then remove the NLOS errors by subtracting the expectation of NLOS errors. This method could mitigate the NLOS propagation effect. Simulation results show that the PKCS has much better performance than those methods without the correction method and significantly improves the localization accuracy.

Acknowledgments

This work was supported by the National High-Tech Research and Development Plan of China under Grant no. 2010AA101702 and the National Natural Science Foundation of China under Grants nos. 60774010 and 60274009.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, The MIT Press, Cambridge, Mass, USA, 2005.

[2] I. Amundson, J. Sallai, X. Koutsoukos, and A. Ledeczi, "Mobile sensor waypoint navigation via RF-based angle of arrival localization," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 842107, 15 pages, 2012.

[3] I. Güvenç and C. C. Chong, "A survey on TOA based wireless localization and NLOS mitigation techniques," *IEEE Communications Surveys and Tutorials*, vol. 11, no. 3, pp. 107–124, 2009.

[4] Y. S. Lee, J. W. Park, and L. Baroll, "A localization algorithm based on AOA for ad-hoc sensor networks," *Mobile Information Systems*, vol. 8, no. 1, pp. 61–72, 2012.

[5] Y. Weng, W. Xiao, and L. Xie, "Total least squares method for robust source localization in sensor networks using TDOA measurements," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 172902, 8 pages, 2011.

[6] J. Wang, Q. H. Gao, Y. Yu, H. Y. Wang, and M. L. Jin, "Toward robust indoor localization based on bayesian filter using chirp-spread-spectrum ranging," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 3, pp. 1622–1629, 2012.

[7] P. C. Chen, "A non-line-of-sight error mitigation algorithm in location estimation," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, pp. 316–320, New Orleans, La, USA, 1999.

[8] S. Nawaz and N. Trigoni, "Robust localization in cluttered environments with NLOS propagation," in *Proceedings of the IEEE 7th International Conference on Mobile Adhoc and Sensor Systems (MASS '10)*, pp. 166–175, San Francisco, Calif, USA, November 2010.

[9] S. Maranò, W. M. Gifford, H. Wymeersch, and M. Z. Win, "NLOS identification and mitigation for localization based on UWB experimental data," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 1026–1035, 2010.

[10] J. Borras, P. Hatrack, and N. B. Mandayam, "Decision theoretic framework for NLOS identification," in *Proceedings of the 48th IEEE Vehicular Technology Conference (VTC '98)*, pp. 1583–1587, May 1998.

[11] S. Mazuelas, F. A. Lago, J. Blas et al., "Prior NLOS measurement correction for positioning in cellular wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 5, pp. 2585–2591, 2009.

[12] L. Cheng, C. D. Wu, and Y. Z. Zhang, "Indoor robot localization based on wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1099–1104, 2011.

[13] H. S. Ahn and W. Yu, "Environmental-adaptive RSSI-based indoor localization," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 4, pp. 626–633, 2009.

[14] A. S. Paul and E. A. Wan, "RSSI-based indoor localization and tracking using sigma-point kalman smoothers," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 5, pp. 860–873, 2009.

[15] M. Saxena, P. Gupta, and B. N. Jain, "Experimental analysis of RSSI-based location estimation in wireless sensor networks," in *Proceedings of the 3rd International Conference on Communication System Software and Middleware*, pp. 503–510, Piscataway, NJ, USA, January 2008.

[16] S. Slijepcevic, S. Megerian, and M. Potkonjak, "Characterization of location error in wireless sensor networks: analysis and applications," in *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks*, pp. 593–608, Palo Alto, Calif, USA, 2003.

[17] M. P. Wylie and J. Holtzman, "The non-line of sight problem in mobile location estimation," in *Proceedings of the 5th IEEE International Conference on Universal Personal Communications*, pp. 827–831, Cambridge, Mass, USA, October 1996.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 962523, 12 pages
doi:10.1155/2012/962523

Review Article

A Survey of Localization in Wireless Sensor Network

Long Cheng,[1] Chengdong Wu,[1] Yunzhou Zhang,[1] Hao Wu,[2] Mengxin Li,[3] and Carsten Maple[4]

[1] *College of Information Science and Engineering, Northeastern University, Shenyang 110819, China*
[2] *Faculty of Engineering, University of Sydney, Sydney, 2006 NSW, Australia*
[3] *Faculty of Information and Control Engineering, Shenyang Jianzhu University, Shenyang 110168, China*
[4] *Department of Computer Science and Technology, University of Bedfordshire, Luton LU1 3JU, UK*

Correspondence should be addressed to Long Cheng, chenglong2000_0@yahoo.com.cn

Received 18 September 2012; Accepted 16 November 2012

Academic Editor: Wei Meng

Localization is one of the key techniques in wireless sensor network. The location estimation methods can be classified into target/source localization and node self-localization. In target localization, we mainly introduce the energy-based method. Then we investigate the node self-localization methods. Since the widespread adoption of the wireless sensor network, the localization methods are different in various applications. And there are several challenges in some special scenarios. In this paper, we present a comprehensive survey of these challenges: localization in non-line-of-sight, node selection criteria for localization in energy-constrained network, scheduling the sensor node to optimize the tradeoff between localization performance and energy consumption, cooperative node localization, and localization algorithm in heterogeneous network. Finally, we introduce the evaluation criteria for localization in wireless sensor network.

1. Introduction

Due to the availability of such low energy cost sensors, microprocessor, and radio frequency circuitry for information transmission, there is a wide and rapid diffusion of wireless sensor network (WSN). Wireless sensor networks that consist of thousands of low-cost sensor nodes have been used in many promising applications such as health surveillance, battle field surveillance, and environmental monitoring. Localization is one of the most important subjects because the location information is typically useful for coverage, deployment, routing, location service, target tracking, and rescue [1]. Hence, location estimation is a significant technical challenge for the researchers. And localization is one of the key techniques in WSN.

The sensor nodes are randomly deployed by the vehicle robots or aircrafts. While the Global Positioning System (GPS) is one of the most popular positioning technologies which is widely accessible, the weakness of high cost and energy consuming makes it different to install in every node. In order to reduce the energy consumption and cost, only a few of nodes which are called beacon nodes contain the GPS modules. The rest of nodes could obtain their locations through localization method. The process of estimating the unknown node position within the network is referred to as node self-localization. And WSN is composed of a large number of inexpensive nodes that are densely deployed in a region of interests to measure certain phenomenon. The primary objective is to determine the location of the target. As shown in Figure 1, we classify the localization method into target/source localization and node self-localization. And the target localization can be further classified into four categories: single-target localization in WSN, multiple-target localization in WSN, single-target localization in wireless binary sensor network (WBSN), and multiple-target localization in WBSN. And node self-localization can be classified into two categories: range-based localization and range-free localization. The former method uses the measured the distance/angle to estimate the location. And the latter method uses the connectivity or pattern matching method to estimate the location. We will present the localization method in some special scenarios and finally introduce the evaluation criteria for localization in WSN.

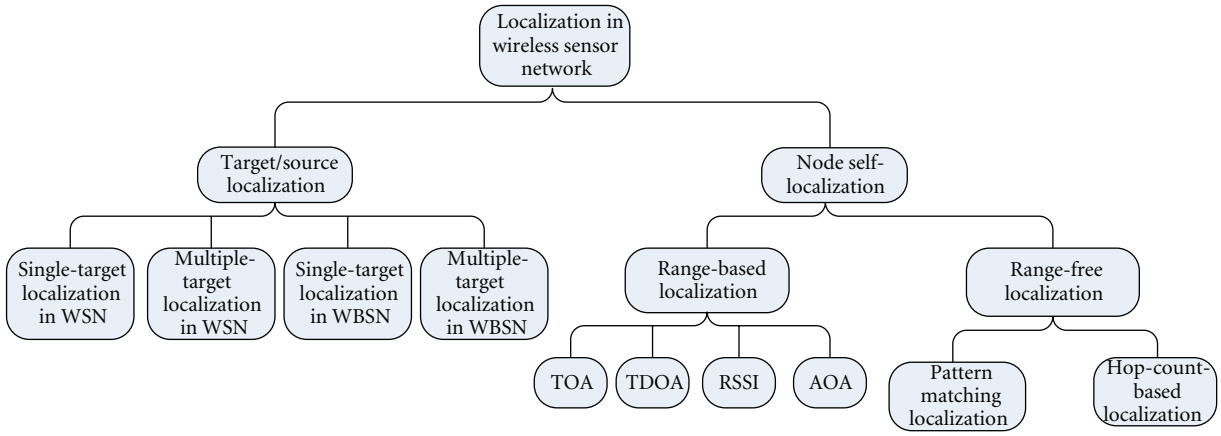


Figure 1: Localization methods taxonomy.

2. Target/Source Localization

2.1. Single-Target/Source Localization in Wireless Sensor Network. The source localization methods have a wide range of possible applications. The outdoor application includes vehicle or aircraft localization. In an indoor environment, this method could track the human speakers. In underwater environment, it can be used to locate the large sea animals and ships. There are several ways to estimate the source location: energy-based, angle of arrival (AOA) [2], time difference of arrival (TDOA) [3–6]. As an inexpensive approach, energy-based method is an attractive method because it requires low hardware configuration. In this survey, we focus on the energy-based source localization.

Single-source localization can be further divided into: energy decay model-based localization algorithm and model-independent localization algorithms.

(1) Decay Model-Based Localization Algorithm. Equation (1) shows the decay model in [7–9]. The received signal strength at ith sensor during time interval t can be written as

$$y_i(t) = g_i \frac{S(t)}{d_{ik}^2(t)} + n_i(t), \tag{1}$$

where g_i represents the gain factor of the ith sensor. We assume that $g_i = 1$. $S(t)$ is the signal energy at 1 meter away. And d_{ik} is the Euclidean distance between the ith sensor and the source. In addition n_i is the measurement noise modeled as zero mean white Gaussian with variance σ_i^2, namely, $n_i \sim N(0, \sigma_i^2)$.

Although this energy decay model appears quite simplistic, it is the one commonly used in the literature. Since the objective function of single-source localization method has multiple local optima and saddle points [7], the authors formulated the problem as a convex feasibility problem and proposed a distributed version of the projection onto convex sets method. A weighted nonlinear least squares and weighted linear least squares methods [8] were proposed to estimate the location of the target. In [9], the authors proposed normalized incremental subgradient algorithm to solve the energy-based sensor network source localization problem where the decay factor of the energy decay method is unknown.

Unlike the signal models in [7–9], the authors derived a more generalized statistical model [10] for energy observation. And a weighted direct/one-step least-squares-based algorithm was investigated to reduce the computational complexity. And in comparison with quadratic elimination method, these methods were amenable to a correction technique which incorporates the dependence of unknown parameters leading to further performance gains. This method offered a good balance between the localization performance and computational complexity. Energy ratio formulation [11] was an alternative approach that is independent of the source energy $S(t)$. This was accomplished by taking ratios of the energy reading of a pair of sensors in the noise-free case. In [12], the authors proposed an energy aware source localization method to reduce the energy consumption in localization.

(2) Model-Independent Methods. A kernel averaging approach [13] which needs not information about energy decay model was proposed. In [14], a novel model-independent localization method was proposed. Since the nodes with higher received signal strength measurement were closer to the source, a distributed sorting algorithm is employed. If the sensor nodes know their rank, the required distance estimates are obtained as the expected value of the respective probability density functions. Finally, the projection onto convex sets (POCS) method was used to estimate the location of the source.

2.2. Multiple-Target Localization in Wireless Sensor Network. Many works investigate the single-target localization. However, very limited papers investigate the multiple-target localization. Most of the works are based on the maximum likelihood estimator. The details of the maximum likelihood estimator are as follows.

The received signal strength at ith sensor during time interval t can be written as

$$y_i(t) = g_i \sum_{k=1}^{K} \frac{S_k(t)}{d_{ik}^{\alpha}(t)} + \varepsilon_i(t), \tag{2}$$

where $d_{ik}(t)$ is the distance between the ith sensor and the kth source. K is the number of the sources. g_i is the gain of ith sensor. $\varepsilon_i(t)$ is random variable with mean μ_i and variance σ_i^2. $S_k(t)$ is the signal energy at 1 meter away for kth source. α is the attenuation exponent.

We define the following matrix notations as follows:

$$Y = \left[\frac{y_1 - \mu_1}{\sigma_1}, \ldots, \frac{y_N - \mu_N}{\sigma_N}\right]^T,$$

$$G = \operatorname{diag}\left[\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_N}\right],$$

$$S = [S_1, S_2, \ldots, S_K]^T,$$

$$D = \begin{bmatrix} \frac{g_1}{d_{11}^2} & \frac{g_1}{d_{12}^2} & \cdots & \frac{g_1}{d_{1K}^2} \\ \frac{g_2}{d_{21}^2} & \frac{g_2}{d_{22}^2} & \cdots & \frac{g_2}{d_{2K}^2} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{g_N}{d_{N1}^2} & \frac{g_N}{d_{N2}^2} & \cdots & \frac{g_N}{d_{NK}^2} \end{bmatrix}, \qquad \varepsilon = [\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N]^N. \tag{3}$$

Using these notations, (2) can be represented as

$$Y = GDS + \varepsilon = HS + \varepsilon, \tag{4}$$

where $H = GD$.

So the joint probability density function (4) can be expressed as

$$f(Y \mid \theta) = (2\pi)^{-N/2} \exp\left(-\frac{1}{2}(Y - HS)^T (Y - HS)\right), \tag{5}$$

where $\theta = [r_1, r_2, \ldots, r_K; S_1, S_2, \ldots, S_K]^T$.

The maximum likelihood estimation is equivalent to minimizing the following function:

$$L(\theta) = (Y - HS)^T (Y - HS) = \|Y - HS\|^2. \tag{6}$$

We can obtain the maximum likelihood parameter estimation of θ by minimizing $L(\theta)$.

To minimize $L(\theta)$, we should take the following operation:

$$\frac{\partial L(\theta)}{\partial S_i} = 0. \tag{7}$$

This condition leads to the following relation:

$S = H^{\nabla} H$, where H^{∇} is the pseudoinverse of the matrix H.

So we get the modified cost function:

$$\arg\min L'(\theta) = \left\| Y - HH^{\nabla} Y \right\|^2. \tag{8}$$

And a multiresolution (MR) search and the expectation maximization (EM) method [15] were proposed to solve (8). An efficient EM algorithm [16] was proposed to improve the estimation accuracy and avoid trapping into local optima through the effective sequential dominant-source initialization and incremental search schemes. An alternating projection [17] algorithm was proposed to decompose the multiple-source localization into a number of simpler, yet also nonconvex, optimization steps. This method could decrease the computation complexity.

2.3. Single-Target/Source Localization in Wireless Binary Sensor Network. Most of the source localization methods are focused on the measured signal strength; that is, the fusion center knows the measurements of the nodes. In order to obtain the measurements, the node needs the complex calculating process. The above methods require transmission of a large amount of data from sensors which may not be feasible under communication constraints. The binary sensors sense signals (infrared, acoustic, light, etc.) from their vicinity, and they only become active by transmitting a signal if the strength of the sensed signal is above a certain threshold. The binary sensor only makes a binary decision (detection or nondetection) regarding the measurement, and consequently, only its ID needs to be sent to the fusion center when it detects the target, otherwise it remains silent. So the binary sensor is a low-power and bandwidth-efficient solution for wireless sensor network.

Limited papers investigate the source localization in binary sensor network. And previous works have been proposed to try to estimate the location of the single source in wireless binary sensor network (WBSN). In [18], the authors proposed a maximum likelihood source location estimator in WBSN. A low complexity source localization method [19] which is based on the intersection of detection areas of sensors was introduced in noisy binary sensor networks. A subtract on negative add on positive (SNAP) [20] algorithm was proposed to identify the source location using the binary sensor networks. This is a fault-tolerant algorithm that is slightly less accurate but it is computationally less demanding in comparison with maximum likelihood estimation. In [21], the authors proposed a trust index based subtract on negative add on positive (TISNAP) method to improve the accuracy of localization for multiple event source localization. This algorithm reduces the impact of faulty nodes on the source localization by decreasing their trust index. And the TISNAP algorithm assumed that the distance between any two sources is far enough; that is, the node is influenced by only one source initially. So the localization process is similar to the single-source localization process. However, all of the previous works mainly focus on single-source localization. Fewer papers investigate the multiple-source localization in WBSN.

3. Node Self-Localization

3.1. Range-Based Localization. The classic methods to estimate the indoor location are time of arrival (TOA), time

difference of arrival (TDOA), angle of arrival (AOA), and received signal strength (RSS). TOA method measures travel times of signals between nodes. TDOA method locates by measuring the signals' arrival time difference between anchor nodes and unknown node. It is able to achieve high ranging accuracy, but requires extra hardware and consumes more energy. As an inexpensive approach, RSS has established the mathematical model on the basis of path loss attenuation with distance [22, 23], and it requires relatively low configuration and energy. We can obtain the distance between the beacon node and unknown node through the above three measurement methods. We set the position of beacon node is $\langle (x_1, y_1), \ldots, (x_N, y_N), \rangle$ and the position of unknown node is $X = [x, y]^T$. $\widetilde{d}_i$ is the estimated distance between ith beacon node and unknown node. We can obtain the coordinate matrix of the unknown node as follows:

$$X = \left(A^T A\right)^{-1} A^T B,$$

$$A = 2\begin{bmatrix} (x_1 - x_2) & (y_1 - y_2) \\ (x_1 - x_3) & (y_1 - y_3) \\ \vdots & \vdots \\ (x_1 - x_{N-1}) & (y_1 - y_{N-1}) \end{bmatrix},$$

$$B = \begin{bmatrix} \widetilde{d}_2^2 - \widetilde{d}_1^2 - (x_2^2 + y_2^2) + (x_1^2 + y_1^2) \\ \widetilde{d}_3^2 - \widetilde{d}_1^2 - (x_3^2 + y_3^2) + (x_1^2 + y_1^2) \\ \vdots \\ \widetilde{d}_{N-1}^2 - \widetilde{d}_1^2 - (x_{N-1}^2 + y_{N-1}^2) + (x_1^2 + y_1^2) \end{bmatrix}. \tag{9}$$

The angles between unknown node and a number of anchor nodes are used in the AOA method to estimate the location. This method needs the antenna array which is an expensive solution for low-cost sensor node.

3.2. Range-Free Localization

3.2.1. Hop-Count-Based Localization. As range-free positioning system, DV-Hop is the typical representation. It does not need to measure the absolute distance between the beacon node and unknown node. It uses the average hop distance to approximate the actual distances and reduces the hardware requirements. It is easy to implement and applicable to large network. But the positioning error is also correspondingly increased.

The positioning process of DV-Hop is divided into three stages: information broadcast, distance calculation, and position estimation. In information broadcast stage, the beacon nodes broadcast their location information package which includes hop count and is initialized to zero for their neighbors. The receiver records the minimal hop of each beacon nodes and ignores the larger hop for the same beacon nodes. Then the receiver increases the hop count by 1 and transmits it to neighbor nodes. All the nodes in a network can record the minimal hop counts of each beacon nodes. In distance calculation stage, according to the position of the beacon node and hop count, each beacon node uses the following equation to estimate the actual distance of every hop:

$$\text{HopSize}_i = \frac{\sum_{j \neq i} \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}}{\sum_{j \neq i} h_j}, \tag{10}$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of beacon nodes i and j, respectively. h_j is the hop count between the beacon nodes. Then, beacon nodes will calculate the average distance and broadcast the information to network. The unknown nodes only record the first average distance and then transmit it to neighbor nodes. Finally, the unknown node calculates its location through (9). In order to improve the localization accuracy, the improved algorithm mainly focuses on the following several aspects: average hop distance between beacon nodes, deployment of the beacon nodes, and node information.

(1) Average Hop Distance between Beacon Nodes. In the randomly deployed node density and connectivity network, Wang et al. [24] proposed a hop progress analytical model to estimate the optimal path distance between any pair of sensor nodes in the network. And it derived an expected hop progress and hop counts estimation method. A range-free localization algorithm (LAEP) which is using the trilateration techniques and the expected hop progress analytical results is proposed. Unlike the DV-Hop method, the LAEP broadcasts the anchor coordinated and the corresponding estimated distance to each sensor at the same time; therefore, it can dramatically reduce the network traffic and the communication delay. Wang only considers the node's receipt of beacon on a line to the utmost extent. Xu et al. [25] proposed a mobile anchor node localization method that is based on Archimedes curve. It takes communication path as curve spread. It avoids the error caused by large straight line dissemination and improves the precision. Lee et al. [26] proposed a robust weighted algorithm which is based on DV-Hop algorithm to calculate the average hop distances between unknown nodes and anchor nodes. It applies to most topological structure networks and reduces the location error. In the same way, Zhang et al. [27] improved the average hop distance based on minimum mean square error standard, which reduced positioning error. Lee et al. [28] used Karush-Kuhn-Tucker (KKT) standards and Lagrange's mean value theorem to correct the average hop distance error and improve location accuracy.

(2) Deployment of the Beacon Nodes. According to the ideal or regular node deployment scheme, the modified DV-Hop method is improved. Zheng et al. [29] firstly derived a beacon nodes deployment strategy that deploys a beacon node in the center of the area and other nodes are equally placed in the circle whose center is the center of the area and radius is half of the length of the area. Based on this deployment strategy, an accurate long-range DV-Hop algorithm is proposed. This method is adapted to large-scale network. Lee et al. [30] put forward a quadratic programming method to optimize

adjacent distance mapping. And it can be applied to the isotropic and anisotropic network.

(3) Node Information. Some modified methods were proposed through the neighbor node information such as anchor information and relationship between node and anchor or topology structure to improve the DV-Hop method. Zhong and He [31] proposed a proximity metric called RSD (regulated signature distance) to capture the distance relationships among 1-hop neighboring nodes. This method can be conveniently applied as a transparent supporting layer for state-of-the-art connectivity-based localization solutions to achieve better accuracy. He et al. [32] proposed a spring swarm localization algorithm (SSLA) which uses the network topology information and a small amount of anchor node location information to calculate unknown nodes position. Chen et al. [33] used the information of neighbor node to make anchor node communication range to be gradient to improve accuracy. Lim and Hou [34] addressed the issue of localization in anisotropic sensor networks. And a linear mapping method is proposed to characterize anisotropic features. It projects one embedding space built upon proximity measures into geographic distance space by using the truncated singular value decomposition (SVD) pseudoinverse technique. This method is different from MDS-Map method and owns higher accuracy than MDS-Map algorithm and the other expanded MDS-Map algorithm.

(4) Comprehensive Improvement Method. In addition to the above aspects, Brida et al. [35] used DV-Hop algorithm, DV-distance, DV-Euclidean algorithm, the constraint, and iteration condition that are to be added to select reference node. Then it used trilateration to find possible unknown node area, the estimated center of area as the final position. This algorithm could reduce the network energy consumption and improves localization accuracy. Chia-Ho [36] put forward a distributed, range-free localization algorithm. In this method, the mobile beacon node with directive antenna is used to supply the location information for the unknown node. Tan et al. [37] exploited acoustic communication to further research underwater range-free algorithm, especially proposing future prospect and development trend in view of the characteristic of underwater acoustic channel.

3.2.2. Pattern Matching Method. Pattern matching localization, also called map-based or Fingerprint algorithm, is one of the most viable solutions for range-free localization methods recently. The fingerprint localization involves two phases. During the first phase, the received signals at selected locations are recorded in an offline database called radio map. Then, the second phase, it works at the online state. The pattern matching algorithms are used to infer the location of unknown node by matching the current observed signal features to the prerecorded values on the map [38, 39].

Fang et al. [40] proposed a novel method to extract the feature of robust signals to efficiently mitigate the multipath effect. This method enhances the robustness under a multipath fading condition and is commonly used for the indoor environments. Swangmuang and Krishnamurthy [41] presented a new analytical model that applies proximity graphs for approximating the probability distribution of error distance, which recorded a location fingerprint database by using the received signals. In addition, there are many problems under the indoor positioning environment as following: how to capture the character of propagation signal in the complex dynamic environment and how to accommodate the receiver gain difference of different mobile devices and so on. Wang et al. [42] solved these problems by modeling them as common mode noise and then developed a location algorithm based on a novel differential radio map. Gogolak et al. [43] proposed fingerprint localization methodology based on neural network which is applied in the real experimental indoor environment. It provided the necessary measurement results to the fingerprint localization.

4. Localization in Some Special Scenarios

Current and potential applications of sensor networks may be quite different. The scale of the network in these applications may be small or large, and the environments may be different. So the traditional localization methods are not suitable for the special scenarios. And there are some challenges for locating sensor nodes that need to be solved. In this survey, we mainly describe the following four challenges. The first challenge is NLOS (non-line-of-sight) ranging error problem. The direct path from the unknown node to the beacon is blocked by obstacles in wireless sensor network; the signal measurements include an error due to the excess path traveled because of the reflection of acoustic signal, which is termed as the NLOS error. The NLOS error results in the large location estimation error. The second challenge is the energy consumption and localization accuracy problem. Since the sensor node is powered by battery, the node may fail due to the depletion of energy. So the energy consumption is critical for the localization problem. It contains the node selection, tradeoff between localization performance and energy consumption, and node resource management. Since the unreliable hardware and complicate communication environment, the received information may be unreliable. Therefore, the third challenge is corporative localization. And the fourth challenge is the localization in heterogeneous sensor network.

4.1. Localization in NLOS Scenario

4.1.1. NLOS Identification/Classification. As shown in Figure 2, there may be no direct path from the beacon node to the unknown node in complicated environment. Because of the reflection and diffraction, the signal which is used for distance measurement can reflect and bound off multiple surfaces before arriving at the receiver. So the signal may actually travel excess path lengths and the direct path is blocked. The signal measurements include an error due to the excess path traveled which is termed as the NLOS error.

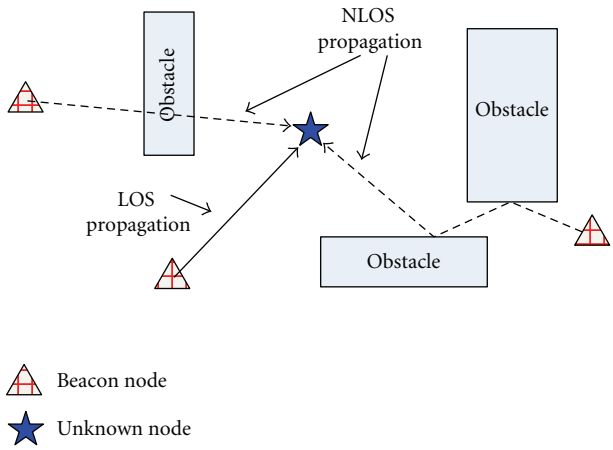


Figure 2: Example of localization in LOS/NLOS environments.

The NLOS problem has been studied in [44] and it was reported that the NLOS error was quite common in all environments except for rural areas. The large location estimation error will occur in NLOS environment. Accordingly, NLOS identification is particularly significant in localization. The problem of NLOS identification is essentially a detection problem. There are two main approaches to solve the NLOS errors: parametric methods and nonparametric methods. In this section, we give a brief overview of some key researches in this area.

Wylie and Holtzman [45] proposed a method based on parameter hypothesis test which determines the measurements whether belong to NLOS by comparing the NLOS variance with the LOS variance. It has a simple criterion but needs the detailed environment parameter and a prior knowledge. Borras et al. [46] investigated NLOS identification by using binary hypothesis test and generalized likelihood ratio for identifying NLOS error. It proposed a proper decision criteria and its premise is that the NLOS error is Gaussian distributed with a large variance. Based on Wylie-Holtzman algorithm, Mazuelas et al. [47] proposed an improvement by using NLOS ratio estimation and it will be able to correct the NLOS measurements from the previous knowledge of this ratio.

The aforementioned methods need a mount of priori knowledge and historical data. Chan et al. [48] proposed a residual test (RT) method to overcome the shortage which needs lots of priori knowledge. It determines the measurements whether belong to NLOS by measuring the samples appropriate to central Chi-distribution. The principle of this method is that if all measurements are LOS, and if the localization technique gives maximum likelihood estimates, then the residuals, normalized by the Cramer Rao Lower Bound (CRLB), will have a central χ^2 distribution. And if the measurements contain the NLOS error, the distribution is noncentral χ^2 distribution. Venkatraman and Caffery Jr. [49] investigated NLOS identification for moving targets by using a time series of range measurements. Gezici et al. [50] proposed a nonparameter-based hypothesis test method which used a distance metric between a known measurement error distribution and a nonparametrically estimated distance measurement distribution. Yu and Guo [51] also proposed a nonparameter-based hypothesis test method by using generalized likelihood ratio to establish the relationship between LOS and NLOS. Then it used Neyman-Pearson (NP) test method for NLOS identification. A sequential probability ratio test [52] which is tolerant to the parameters fluctuations is employed to identify whether the measurement contains the non-line-of-sight (NLOS) errors.

4.1.2. NLOS Mitigation. Because of the existence of the non-ideal channel condition and non-line-of-sight transmission between the unknown nodes and beacon nodes, NLOS error mitigation has become a key technology and hotspot in the research about location estimation in wireless sensor network.

The first way attempts to identify the propagation conditions (LOS or NLOS) and then eliminate the measurements in NLOS; they only use the measurements in LOS to locate the unknown node. The propagation model-based method [53, 54] either directly employs the existing propagation models or empirically develops a model based on experimental results. The second way uses all NLOS and LOS measurements to estimate the location, but provides weighting or sealing to minimize the effects of the NLOS contributions. The weighting is determined by either the position geometry and beacon nodes layout or the residuals (fitting errors) of individual beacon node. The Taylor series linearization [55] (TS-LS), a widely used localization algorithm, should have the prior information of the error statistics which can be used to determine the weights. Chen [56] develops an algorithm to mitigate the NLOS errors by residual weighting when the range measurements corrupted by NLOS errors are not identifiable. The hypothesis testing [57] is employed to detect whether the environment is NLOS or LOS along with time of arrival (TOA) and received signal strength (RSS) measurements. And then an extended Kalman filter is used to nonlinear estimation.

In a scattering environment, most of the propagation paths between the unknown nodes and beacon nodes are NLOS; the constrained optimization techniques are used to reduce NLOS errors [58]. The neural network is employed to predict the NLOS error [59]; Kalman filters [60] and modified two-stage Kalman filter [61] are used to correct NLOS measurements.

All the positioning algorithms in NLOS environment focused more on the cellular network. These methods could be used in some scenarios for wireless sensor network localization. Fewer methods investigate the NLOS mitigation algorithm for Wireless sensor network.

4.2. Node Selection Criteria for Localization in Energy-Constrained Network. Due to the limited power of sensor node and hostile deployment environment, the node selection in WSN is different from the traditional node selection in traditional wireless network. If all the sensor nodes are used at the same time to executive localization

task without selection, although the energy consumption of nodes selection are saved, but at this time the repeatability of the received information would be quite larger. If the nodes are random selected to executive the localization task, the algorithm is simple and the extra overhead can be ignored, but localization accuracy is low in this case. Obviously, this method cannot satisfy the user's requirements for the accurate localization, the unbalance of energy consumption will appear, and some nodes may fail due to the depletion of energy. This may affect the network connectivity and may result in losing the sensed data. These characteristics of WSN determine selection method which is different from traditional network. Therefore, it is necessary to investigate nodes' selection mechanism in WSN.

The primary algorithm makes decision with global information [62]; this method minimized the expected filtered mean-squared position error for a given number of active nodes by using a global knowledge of all node locations. This algorithm needs the positions of nodes and broadcasts them to all nodes and a lot of data communication; therefore, it only can be applied to small networks. Based on the former algorithm, a local selection strategy is investigated [63]. This method determines whether or not that node should be active by only incorporating geometrical knowledge of itself and the active set of nodes from the previous information. Based on this approach, the researchers have also investigated other strategies, such as the least square method, Bayes probability method [64, 65]. Furthermore, in order to narrow the scope and scale of selected nodes, researchers proposed a method which combines the track and the current state of the robot. Zhang and Cao [66] proposed a multinode cooperation dynamic tree algorithm. This method ensured that spanning tree has low energy consumption and high information content by increasing and decreasing the number of the nodes dynamically. But this method still had some disadvantages: the root node needs data fusion and the new node needs to be calculated, and the consumption of energy is quite higher. Yang et al. [67] proposed online prediction based on particle filter and estimate the probability distribution of the target state under the Bayes framework. This method realized the optimal selection of the node sequence and introduced a shortest path algorithm to reduce the information transmission. Hamouda and Phillips [68] proposed a method which employs the moving speed of the mobile robot to improve the localization accuracy and consistency.

The signal shielding and multipath interference make the channel parameters become too complex to definite error factor. Bel et al. [69] proposed two selection principles to reduce the number of active nodes, and the nodes with accurate measured value (RSSI value larger than specified threshold) are selected. This method is effective to balance the accuracy and energy consumption and is suitable for the WSN which is hardware resource constrained. Zhao and Nehorai [70] used the Cramer-Rao equation to select the next node to participate in positioning. Because of the complexity of the observed model and the non-Gaussian noise, it is hard to get the optimal solution of the problem.

4.3. Scheduling the Sensor Node to Optimize the Tradeoff between Localization Performance and Energy Consumption. A typical sensor network consists of a large number of small sensors which are deployed randomly. However, a sensor node has limited resources because of battery power and small memory. Therefore, nodes' resource management is compulsory. In typical sensor network applications, nodes are deployed in an unattended environment such as disaster management, habitat monitoring, industrial process control, and object tracking. Enormous event data will be generated for a long sensing time in WSN. Hence, by the methods of nodes resource management, effective usage of the vast amount of data is crucial. In the meanwhile, the scalability of both energy and spatial dimensions in distributed sensor network is a key issue. Sensor networks must track various phenomena at the same time and work within limited communication bandwidth, energy, and processing speed. Therefore, it is critical to distribute the workload across only the "relevant" sensors equally and leave other sensors available for other things. These characteristics of WSN determine the importance of nodes resource management.

Energy consumption is one of the most important issues in recent years. Ren and Meng [71] proposed a localization algorithm based on particle filtering for sensor networks. Assisted by multiple-transmit-power information, it outperforms the existing algorithms that do not utilize multiple-power information. You et al. [72] proposed a specified positional error tolerance, the sensor-enhanced and energy-efficient adaptive localization system in an application. This localization system dynamically sets sleep time for the nodes and adapting the sampling rate of target's mobility level. However, the process of error estimation dynamically relies on several factors in the specific environment. Gribben et al. [73] proposed a scheduling algorithm that selects a subset of active beacon nodes to be used in localization. It served to reduce the message overhead, increased network lifetime, and improved localization accuracy in dense mobile networks. However, maximizing the nodes' sleep time is much more energy efficient if the nodes never wake up until the reception of wake-up messages. The above algorithms have the same feature that the duty cycle of the sensor nodes is fixed in advance. In [74], the authors proposed an innovative probabilistic wake-up protocol for energy-efficient event detection in WSNs. The main idea of it is to reduce the duty cycle of every sensor via probabilistic wake-up through the dense deployment of sensor networks.

The problem of unique network localization and a mathematical topic known as rigidity theory have a strong connection. Goldenberg et al. [75] proposed a localization method for sparse networks by sweeping techniques. This method is saving all possible positions in each position step and pruning incompatible ones. One drawback of sweeping method is that the possible positions could increase exponentially as long as the number of nodes increased. Other types of localization methods are also available, such as using multidimensional scaling [76, 77] or mobile anchors [78, 79]. However, all the previous works tried to localize

more sensor nodes in a network without guaranteeing all of them. Khan et al. [80] introduced a localization method to localize all nodes by the minimal number of anchor nodes. However, they assume that the sensing range of each sensor can be enlarged to guarantee certain triangulation, so that three anchor nodes are enough to localize all sensors.

4.4. Cooperative Node Localization. There may be not enough information in the concentrated network or the node may contain the harmful information in sparse network. There are two branches in this area: (1) access the accuracy and reliability of the neighborhood nodes. (2) Improve precision with the cooperation of the active and passive nodes.

Some nodes may bring unreliable or even harmful information [81], so it is essential to review the received information. Tam et al. [82] employed the nearest link as reference to review the information. When there are massive link in dense network and positioning mainly depends on the geometry of the neighbor node topology information, the nearest neighbors may not correspond to the best link. Aiming at this issue, Denis et al. [83] proposed an adaptive method to eliminate the inefficient links, but this method has to work with neighbor node information, and the method cannot effectively reduce the number of packet effectively. Therefore, Das and Wymeersch [84] put forward a kind of distributed criterion; this method employed Cramer-Rao limit as identifiable parameters to identify the links. This method could avoid the invalid neighbor node links and unreliable transmission; thus, it can effectively reduce the computation time and the number of packets.

The accuracy of master-slave node cooperative localization is mainly depended on the measurements accuracy and the number of primary reference nodes (PRN, Primary Reference Node). But in the actual application, it is difficult to increase the number of primary reference nodes because of the factors of energy and the complexity. Wymeersch et al. [81] and Fujiwara et al. [85] put forward a new method: the nodes which received the information of the target and the primary reference nodes are termed as secondary reference nodes (SRN, secondary Reference Node), the SRNs participated in the localization in a passive way. This cooperative method reduced the required number of PRN with relatively higher localization accuracy. Gholami et al. [86] used the maximum likelihood estimation method to obtain the target position. The authors formulated the localization problem into finding the intersection of the vertex set by using geometry description. This method avoids getting into the local optimum.

4.5. Localization Algorithm in Heterogeneous Sensor Network. Most of the localization methods for the wireless sensor networks are only to consider the homogeneous network. The different kinds of the nodes such as the different maximum communication radius and the different nodes own the different localization mechanisms are not considered in homogeneous, so the localization methods for the homogeneous network cannot be directly applied in the heterogeneous wireless sensor networks.

Du et al. [87] propose a new boundary nodes localization method by using a small number of anchor nodes. First the boundary nodes are elected and their positions are determined. Then the location information of boundary nodes is sent to other nodes through a small hop communication range. Finally, other nodes estimate their locations by the hop count and hop range. The scheme uses fewer beacon nodes, but has much smaller localization error and standard deviation. This method uses fewer beacon nodes, but with a smaller location error and standard deviation. Dong et al. [88] proposed a two-step localization method for two-tiered hierarchical heterogeneous sensor networks. The network consists of three types of nodes: anchor nodes with known locations, a few nodes equipped with both Ultrawide Band (UWB) and RF radios, and a large number of normal sensor nodes. The localization method works in two steps: firstly the high-accurate ranging capability of UWB nodes is used to estimate their location from a few anchor nodes, then, sensor nodes estimate their locations by using UWB nodes as anchor nodes.

Sometimes the distance between some nodes can be measured directly, while others cannot be. Selecting a different positioning algorithms accord to the mutual distance between nodes can be measured or not. Chiang et al. [89] proposed a hybrid unified Kalman tracking (HUKT) technique. The accuracy of tracking is based on both time of arrival (TOA) and time difference of arrival (TDOA) measurements. This method is proposed to adaptively adjust the weighting value between the TOA and TDOA measurements. The scheme can both provide higher localization accuracy for mobile network and adapt to environments with insufficient signal sources.

According to the different communication radius of the nodes, some super nodes can be deployed at some areas with plenty communication demands to transmit the information. Shen and Pesch [90] considered the nodes with more power and longer communication range as the heterogeneous nodes and propose a heuristic relay positioning algorithm for heterogeneous wireless sensor networks, to achieve the sharing of resources in heterogeneous wireless sensor network by using the relay nodes.

5. Evaluation Criteria for Localization in Wireless Sensor Network

The localization errors are inevitable in the estimations. In this section, we describe some common metrics: average localization error, root mean square error, and geometric mean error. And the Euclidean distance and Manhattan distance are two widely used metrics that are computed considering a two-dimensional coordinate system [91]. The Euclidean distance is defined to be the shortest distance between two coordinates. The Manhattan distance is defined to be the distance between two coordinates measured along

the axes at the right angles. The metrics are described as follows.

(1) Average Localization Error. The average localization error for Euclidean distance can be computed as follows:

$$\text{error} = \frac{1}{N_t}\sum_{i=1}^{N_t}\sqrt{(\hat{x}_i - x)^2 + (\hat{y}_i - y)^2}, \tag{11}$$

where N_t is the number of trails. (x, y) is the true location of the unknown node or source. $(\hat{x}_i, \hat{y}_i)$ is the estimated location.

The average localization error for Manhattan distance can be computed as follows:

$$\text{error} = \frac{1}{N_t}\sum_{i=1}^{N_t}(|\hat{x}_i - x| + |\hat{y}_i - y|). \tag{12}$$

(2) Root Mean Square Error. The root mean square error for Euclidean distance can be computed as follows:

$$\text{error} = \sqrt{\frac{1}{N_t}\sum_{i=1}^{N_t}\left((\hat{x}_i - x)^2 + (\hat{y}_i - y)^2\right)}. \tag{13}$$

The root mean square error for Manhattan distance can be computed as follows:

$$\text{error} = \sqrt{\frac{1}{N_t}\sum_{i=1}^{N_t}(|\hat{x}_i - x| + |\hat{y}_i - y|)}. \tag{14}$$

(3) Geometric Mean Error. The geometric mean error for Euclidean distance can be computed as follows:

$$\text{error} = \sqrt[N_t]{\prod_{i=1}^{N_t}\left((\hat{x}_i - x)^2 + (\hat{y}_i - y)^2\right)}. \tag{15}$$

The geometric mean error for Manhattan distance can be computed as follows:

$$\text{error} = \sqrt[N_t]{\prod_{i=1}^{N_t}(|\hat{x}_i - x| + |\hat{y}_i - y|)}. \tag{16}$$

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant nos. 61203216, 61273078) and the Fundamental Research Fund for the Central Universities of China (N110404030, N110804004, and N110404004).

References

[1] C. Wang and L. Xiao, "Sensor localization in concave environments," *ACM Transactions on Sensor Networks*, vol. 4, no. 1, article 3, 2008.

[2] L. M. Kaplan, Q. Le, and P. Molnar, "Maximum likelihood methods for bearings-only target localization," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 3001–3004, Salt Lake City, Utah, USA, May 2001.

[3] Y. Weng, W. Xiao, and L. Xie, "Total least squares method for robust source localization in sensor networks using TDOA measurements," *International Journal of Distributed Sensor Networks*, vol. 2011, Article ID 172902, 8 pages, 2011.

[4] X. Qu and L. Xie, "Source localization by TDOA with random sensor position errors—part I: static sensors," in *Proceedings of the 15th International Conference on Information Fusion*, pp. 48–53, Singapore, July 2012.

[5] X. Qu and L. Xie, "Source localization by TDOA with random sensor position errors—part II: mobile sensors," in *Proceedings of the 15th International Conference on Information Fusion*, pp. 54–59, Singapore, July 2012.

[6] K. C. Ho, "Bias reduction for an explicit solution of source localization using TDOA," *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2101–2114, 2012.

[7] D. Blatt and A. O. Hero, "Energy-based sensor network source localization via projection onto convex sets," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3614–3619, 2006.

[8] K. Deng and Z. Liu, "Weighted least-squares solutions of energy-based collaborative source localization using acoustic array," *International Journal of Computer Science and Network Security*, vol. 7, no. 1, pp. 159–165, 2007.

[9] Q. Shi and C. He, "A new incremental optimization algorithm for ML-based source localization in sensor networks," *IEEE Signal Processing Letters*, vol. 15, pp. 45–48, 2008.

[10] C. Meesookho, U. Mitra, and S. Narayanan, "On energy-based acoustic source localization for sensor networks," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 365–377, 2008.

[11] D. Li and Y. H. Hu, "Energy-based collaborative source localization using acoustic microsensor array," *EURASIP Journal on Advances in Signal Processing*, vol. 2003, article 985029, 2003.

[12] E. Maşazade, R. Niu, P. K. Varshney, and M. Keskinoz, "Energy aware iterative source localization for wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 58, no. 9, pp. 4824–4835, 2010.

[13] M. G. Rabbat, R. D. Nowak, and J. Bucklew, "Robust decentralized source localization via averaging," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 5, pp. V1057–V1060, Philadelphia, Pa, USA, March 2005.

[14] D. Ampeliotis and K. Berberidis, "Energy-based model-independent source localization in wireless sensor networks," in *Proceedings of the 16th European Signal Processing Conference*, Lausanne, Switzerland, August 2008.

[15] X. Sheng and Y. H. Hu, "Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 53, no. 1, pp. 44–53, 2005.

[16] W. Meng, W. Xiao, and L. Xie, "An efficient EM algorithm for energy-based multisource localization in wireless sensor networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 3, pp. 1017–1027, 2011.

[17] D. Ampeliotis and K. Berberidis, "Low complexity multiple acoustic source localization in sensor networks based on energy measurements," *Signal Processing*, vol. 90, no. 4, pp. 1300–1312, 2010.

[18] R. Niu and P. Varshney, "Target location estimation in wireless sensor networks using binary data," in *Proceedings of the 38th International Conference on Information Sciences and Systems*, pp. 17–19, Princeton, NJ, USA, March 2004.

[19] X. Liu, G. Zhao, and X. Ma, "Target localization and tracking in noisy binary sensor networks with known spatial topology," *Wireless Communications and Mobile Computing*, vol. 9, no. 8, pp. 1028–1039, 2009.

[20] M. P. Michaelides and C. G. Panayiotou, "SNAP: fault tolerant event location estimation in sensor networks using binary data," *IEEE Transactions on Computers*, vol. 58, no. 9, pp. 1185–1197, 2009.

[21] X. Xu, X. Gao, J. Wan, and N. Xiong, "Trust index based fault tolerant multiple event localization algorithm for WSNs," *Sensors*, vol. 11, no. 7, pp. 6555–6574, 2011.

[22] K. Lu, X. Xiang, D. Zhang, R. Mao, and Y. Feng, "Localization algorithm based on maximum a posteriori in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 260302, 7 pages, 2012.

[23] L. Cheng, C. D. Wu, Y. Z. Zhang, and Y. Wang, "Indoor robot localization based on wireless sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 3, pp. 1099–1104, 2011.

[24] Y. Wang, X. Wang, D. Wang, and D. P. Agrawal, "Range-free localization using expected hop progress in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 10, pp. 1540–1552, 2009.

[25] H. Xu, Y. Tu, W. Xiao, Y. Mao, and T. Shen, "An archimedes curve-based mobile anchor node localization algorithm in wireless sensor networks," in *Proceedings of the 8th World Congress on Intelligent Control and Automation (WCICA '10)*, pp. 6993–6997, Jinan, China, July 2010.

[26] J. Lee, W. Chung, and E. Kim, "Robust DV-Hop algorithm for localization in wireless sensor network," in *Proceedings of the International Conference on Control, Automation and Systems*, pp. 2506–2509, Gyeonggi-do, South Korea, October 2010.

[27] J. Zhang, W. Li, D. Cui, X. Sun, and F. Zhou, "Study on improved DV-Hop node localization algorithm in wireless sensor network," in *Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications (ICIEA '10)*, pp. 1855–1858, Taichung, Taiwan, June 2010.

[28] S. W. Lee, D. Y. Lee, and C. W. Lee, "Enhanced DV-Hop algorithm with reduced hop-size error in ad hoc networks," *IEICE Transactions on Communications*, vol. 94, no. 7, pp. 2130–2132, 2011.

[29] Y. Zheng, L. Wan, Z. Sun, and S. Mei, "A long range DV-Hop localization algorithm with placement strategy in wireless sensor networks," in *Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '08)*, Dalian, China, October 2008.

[30] J. Lee, W. Chung, and E. Kim, "A new range-free localization method using quadratic programming," *Computer Communications*, vol. 34, no. 8, pp. 998–1010, 2011.

[31] Z. Zhong and T. He, "RSD: a metric for achieving range-free localization beyond connectivity," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 11, pp. 1943–1951, 2011.

[32] Q. B. He, F. Chen, S. Cai, J. Hao, and Z. Liu, "An efficient range-free localization algorithm for wireless sensor networks," *Science China Technological Sciences*, vol. 54, no. 5, pp. 1053–1060, 2011.

[33] Y. Chen, W. Chung, and S. Yuan, "Order-based localization scheme for ad hoc sensor networks," in *Proceedings of the 73rd IEEE Vehicular Technology Conference*, pp. 1–5, Budapest, Hungary, May 2011.

[34] H. Lim and J. C. Hou, "Distributed localization for anisotropic sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 2, article 11, 2009.

[35] P. Brida, J. Machaj, and J. Duha, "A novel optimizing algorithm for DV based positioning methods in ad hoc networks," *Elektronika ir Elektrotechnika*, no. 1, pp. 33–38, 2010.

[36] O. Chia-Ho, "A localization scheme for wireless sensor networks using mobile anchors with directional antennas," *IEEE Sensors Journal*, vol. 7, no. 11, pp. 1607–1616, 2011.

[37] H. P. Tan, R. Diamant, and W. K. G. Seah, "A survey of techniques and challenges in underwater localization," *Ocean Engineering*, vol. 38, no. 14-15, pp. 1663–1676, 2011.

[38] X. Luo, W. J. O'Brien, and C. L. Julien, "Comparative evaluation of Received Signal-Strength Index (RSSI) based indoor localization techniques for construction jobsites," *Advanced Engineering Informatics*, vol. 25, no. 2, pp. 355–363, 2011.

[39] D. J. Suroso, P. Cherntanomwong, P. Sooraksa, and J. Takada, "Fingerprint-based technique for indoor localization in wireless sensor networks using Fuzzy C-Means clustering algorithm," in *Proceedings of the International Symposium on Intelligent Signal Processing and Communications Systems*, pp. 1–5, Chiang Mai, Thailand, December 2011.

[40] S. H. Fang, T. N. Lin, and K.-C. Lee, "A novel algorithm for multipath fingerprinting in indoor WLAN environments," *IEEE Transactions on Wireless Communications*, vol. 7, no. 9, pp. 3579–3588, 2008.

[41] N. Swangmuang and P. Krishnamurthy, "An effective location fingerprint model for wireless indoor localization," *Pervasive and Mobile Computing*, vol. 4, no. 6, pp. 836–850, 2008.

[42] J. Wang, Q. Gao, H. Wang et al., "Differential radio map-based robust indoor localization," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, article 17, 2011.

[43] L. Gogolak, S. Pletl, and D. Kukolj, "Indoor fingerprint localization in WSN environment based on neural network," in *Proceedings of the 9th IEEE International Symposium on Intelligent Systems and Informatics*, pp. 293–296, Subotica, Serbia, September 2011.

[44] M. I. Silventoinen and T. Rantalainen, "Mobile station emergency locating in GSM," in *Proceedings of IEEE International Conference on Personal Wireless Communications*, pp. 232–238, New Delhi, India, February 1996.

[45] M. P. Wylie and J. Holtzman, "The non-line of sight problem in mobile location estimation," in *Proceedings of the 5th IEEE International Conference on Universal Personal Communications Record (ICUPC '96)*, pp. 827–831, Cambridge, Mass, USA, October 1996.

[46] J. Borras, P. Hatrack, and N. B. Mandayam, "Decision theoretic framework for NLOS identification," in *Proceedings of the 48th IEEE Vehicular Technology Conference (VTC '98)*, pp. 1583–1587, Ottawa, Canada, May 1998.

[47] S. Mazuelas, F. A. Lago, J. Blas et al., "Prior NLOS measurement correction for positioning in cellular wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 5, pp. 2585–2591, 2009.

[48] Y. T. Chan, W. Y. Tsui, H. C. So, and P. C. Ching, "Time-of-arrival based localization under NLOS conditions," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 1, pp. 17–24, 2006.

[49] S. Venkatraman and J. J. Caffery, "Statistical approach to non-line-of-sight BS identification," in *Proceedings of the*

5th International Symposium on Wireless Personal Multimedia Communications, vol. 1, pp. 296–300, Honolulu, Hawaii, USA, October 2002.

[50] S. Gezici, H. Kobayashi, and H. V. Poor, "Non-parametric non-line-of-sight identification," in *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC '03-Fall)*, vol. 4, pp. 2544–2548, Orlando, Fla, USA, October 2003.

[51] K. Yu and Y. J. Guo, "Statistical NLOS identification based on AOA, TOA, and signal strength," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 1, pp. 274–286, 2009.

[52] L. Cheng, C. D. Wu, Y. Z. Zhang, and Y. Wang, "An indoor localization strategy for mini-UAV in presence of obstacles," *International Journal of Advanced Robotic Systems*, vol. 2012, pp. 1–8, 2012.

[53] S. Al-Jazzar, J. Caffery Jr., and H. R. You, "A scattering model based approach to NLOS mitigation in TOA location systems," in *Proceedings of the 55th IEEE Vehicular Technology Conference*, pp. 861–865, Birmingham, Ala, USA, May 2002.

[54] L. Liu, P. Deng, and P. Fan, "A TOA reconstruction method based on ring of scatterers model," in *Proceedings of the 4th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT '03)*, pp. 375–377, Chengdu, China, August 2003.

[55] W. H. Foy, "Position-location solutions by Taylor-series estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 12, no. 2, pp. 187–194, 1976.

[56] P. C. Chen, "A non-line-of-sight error mitigation algorithm in location estimation," in *Proceedings of IEEE Wireless Communications and Networking Conference*, vol. 1, pp. 316–320, New Orleans, La, USA, September 1999.

[57] L. Cheng, C. D. Wu, Y. Z. Zhang, and H. Chu, "Mobile location estimation scheme in NLOS environment," *IEICE Electronics Express*, vol. 8, no. 21, pp. 1829–1835, 2011.

[58] S. Venkatraman, J. J Caffery, and H. R. You, "Location using LOS range estimation in NLOS environments," in *Proceedings of the 55th Vehicular Technology Conference*, pp. 856–860, Birmingham, Ala, USA, May 2002.

[59] I. Povescu, I. Nafomita, P. Constantinou, A. Kanatas, and N. Moraitis, "Neural networks applications for the prediction of propagation path loss in urban environments," in *Proceedings of the 53rd IEEE Semi-Annual Vehicular Technology Conference*, pp. 387–391, Rhodes, Greece, May 2001.

[60] B. L. Le, K. Ahmed, and H. Tsuji, "Mobile location estimator with NLOS mitigation using Kalman filtering," in *Proceedings of IEEE Wireless Communications and Networking (WCNC '03)*, vol. 3, pp. 1969–1973, New Orleans, La, USA, March 2003.

[61] W. Kim, G. I. Jee, and J. Lee, "Wireless location with NLOS error mitigation in Korean CDMA system," in *Proceedings of the 2nd International Conference on 3G Mobile Communication Technologies*, pp. 134–138, London, UK, March 2001.

[62] L. M. Kaplan, "Global node selection for localization in a distributed sensor network," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 113–135, 2006.

[63] L. M. Kaplan, "Local node selection for localization in a distributed sensor network," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 136–146, 2006.

[64] A. Bel, J. L. Vicario, and G. Seco-Granados, "Node selection for cooperative localization: efficient energy vs. accuracy tradeoff," in *Proceedings of the 5th IEEE International Symposium on Wireless Pervasive Computing (ISWPC '10)*, pp. 307–312, Modena, Italy, May 2010.

[65] X. J. Yang, K. Y. Xing, K. L. Shi, and Q. Pan, "Dynamic collaborative algorithm for maneuvering target tracking in sensor networks," *Acta Automatica Sinica*, vol. 33, no. 10, pp. 1029–1035, 2007.

[66] W. S. Zhang and G. H. Cao, "DCTC: dynamic convoy tree-based collaboration for target tracking in sensor networks," *IEEE Transactions on Wireless Communications*, vol. 3, no. 5, pp. 1689–1701, 2004.

[67] X. J. Yang, K. Y. Xing, K. L. Shi, and Q. Pan, "Dynamic collaborative algorithm for maneuvering target tracking in sensor networks," *Acta Automatica Sinica*, vol. 33, no. 10, pp. 1029–1035, 2007.

[68] Y. E. M. Hamouda and C. Phillips, "Metadata based optimal sensor selection for multi-target tracking in wireless sensor networks," *International Journal of Research and Reviews in Computer Science*, vol. 2, no. 1, pp. 189–200, 2011.

[69] A. Bel, J. Lopez Vicario, and G. Seco-Granados, "Real-time path loss and node selection for cooperative localization in wireless sensor networks," in *Proceedings of the 21st IEEE International Symposium on Personal, Indoor and Mobile Radio Communications Workshops (PIMRC '10)*, pp. 283–288, Istanbul, Turkey, September 2010.

[70] T. Zhao and A. Nehorai, "Information-driven distributed maximum likelihood estimation based on Gauss-Newton method in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 55, no. 9, pp. 4669–4682, 2007.

[71] H. Ren and M. Q. Meng, "Power adaptive localization algorithm for wireless sensor networks using particle filter," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 5, pp. 2498–2508, 2009.

[72] C. You, Y. Chen, J. Chiang, P. Huang, H. Chu, and S. Lau, "Sensor-enhanced mobility prediction for energy-efficient localization," in *Proceedings of the 3rd Annual IEEE Conference on Sensor and Ad Hoc Communications and Networks*, vol. 1, pp. 565–574, Reston, Va, USA, September 2006.

[73] J. Gribben, A. Boukerche, and R. Pazzi, "Scheduling for scalable energy-efficient localization in mobile ad hoc networks," in *Proceedings of the 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '10)*, Boston, Mass, USA, June 2010.

[74] Y. Zhu and L. M. Ni, "Probabilistic wakeup: adaptive duty cycling for energy efficient event detection," in *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM '07)*, pp. 360–367, Chania, Greece, October 2007.

[75] D. K. Goldenberg, P. Bihler, M. Cao et al., "Localization in sparse networks using sweeps," in *Proceedings of the 12th Annual International Conference on Mobile Computing and Networking (MOBICOM '06)*, pp. 110–121, Los Angeles, Calif, USA, September 2006.

[76] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 4, pp. 2652–2661, Hong Kong, China, March 2004.

[77] Y. Shang and W. Ruml, "Improved MDS-based localization," in *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '04)*, vol. 4, pp. 2640–2651, Hong Kong, China, March 2004.

[78] K.-F. Ssu, C.-H. Ou, and H. C. Jiau, "Localization with mobile anchor points in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 3, pp. 1187–1197, 2005.

[79] M. Erol, L. F. M. Vieira, and M. Gerla, "Localization with Dive'n'Rise (DNR) beacons for underwater acoustic sensor

networks," in *Proceedings of the 2nd Workshop on Underwater Networks (WuWNet '07)*, pp. 97–100, Montreal, Canada, September 2007.

[80] U. A. Khan, S. Kar, and J. M. F. Moura, "Distributed sensor localization in random environments using minimal number of anchor nodes," *IEEE Transactions on Signal Processing*, vol. 57, no. 5, pp. 2000–2016, 2009.

[81] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, 2009.

[82] V. Tam, K. Cheng, and K. Lui, "Using micro-genetic algorithms to improve localization in wireless sensor networks," *Journal of Communications*, vol. 1, no. 4, pp. 1–10, 2006.

[83] B. Denis, M. Maman, and L. Ouvry, "On the scheduling of ranging and distributed positioning updates in cooperative IR-UWB networks," in *Proceedings of IEEE International Conference on Ultra-Wideband (ICUWB '09)*, pp. 370–375, Vancouver, Canada, September 2009.

[84] K. Das and H. Wymeersch, "Censored cooperative positioning for dense wireless networks," in *Proceedings of the 21st IEEE International Symposium on Personal, Indoor and Mobile Radio Communications Workshops (PIMRC '10)*, pp. 262–266, Istanbul, Turkey, September 2010.

[85] R. Fujiwara, K. Mizugaki, T. Nakagawa, D. Maeda, and M. Miyazaki, "TOA/TDOA hybrid relative positioning system using UWB-IR," in *Proceedings of IEEE Radio and Wireless Symposium (RWS '09)*, pp. 679–682, San Diego, Calif, USA, January 2009.

[86] M. R. Gholami, S. Gezici, M. Rydström, and E. G. Ström, "A distributed positioning algorithm for cooperative active and passive sensors," in *Proceedings of the 21st IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC '10)*, pp. 1713–1718, Instanbul, Turkey, September 2010.

[87] X. Du, D. Mandala, W. Zhang, C. You, and Y. Xiao, "A boundary-node based localization scheme for heterogeneous wireless sensor networks," in *Proceedings of IEEE Military Communications Conference (MILCOM '07)*, pp. 1–7, Orlando, Fla, USA, October 2007.

[88] S. Dong, P. Agrawal, and K. Sivalingam, "Localization error evaluation in heterogeneous sensor networks," in *Proceedings of IEEE Global Telecommunications Conference*, pp. 1–5, New Orleans, La, USA, December 2008.

[89] C. T. Chiang, P. H. Tseng, and K. T. Feng, "Hybrid unified kalman tracking algorithms for heterogeneous wireless localization systems," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 2, pp. 702–715, 2012.

[90] C. Shen and D. Pesch, "A heuristic relay positioning algorithm for heterogeneous wireless networks," in *Proceedings of the 69th IEEE Vehicular Technology Conference*, pp. 1–5, Barcelona, Spain, April 2009.

[91] H. Aksu, D. Aksoy, and I. Korpeoglu, "A study of localization metrics: evaluation of position errors in wireless sensor networks," *Computer Networks*, vol. 55, no. 15, pp. 3562–3577, 2011.

Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 294578, 8 pages
doi:10.1155/2012/294578

Research Article

A Minimax Unbiased Estimation Fusion in Distributed Multisensor Localization and Tracking

Xiaomei Qu[1] and Jie Zhou[2]

[1] *College of Computer Science and Technology, Southwest University for Nationalities, Chengdu, Sichuan 610041, China*
[2] *College of Mathematics, Sichuan University, Chengdu, Sichuan 610064, China*

Correspondence should be addressed to Jie Zhou, jzhou@scu.edu.cn

Received 30 August 2012; Accepted 26 October 2012

Academic Editor: Long Cheng

A minimax estimation fusion in distributed multisensor systems is proposed, which aims to minimize the worst-case squared estimation error when the cross-covariances between local sensors are unknown and the normalized estimation errors of local sensors are norm bounded. The proposed estimation fusion is called as the Chebyshev fusion estimation (CFE) because its geometrical interpretation is in coincidence with the Chebyshev center, which is a nonlinear combination of local estimates. Theoretically, the CFE is better than any local estimator in the sense of the worst-case squared estimation error and is robust to the choice of the supporting bound. The simulation results illustrate that the proposed CFE is a robust fusion in localization and tracking and more accurate than the previous covariance intersection (CI) method.

1. Introduction

Multi-sensor networks have received an increasing attention in recent years, due to their huge potential in applications, such as communication, signal process, routing and sensor management, and many other areas. In this paper, we focus on a specific and simple estimation fusion model in a distributed multi-sensor system, which is in fact a two-level optimization in the estimation fusion. Every sensor first optimally estimates the state of target based on its own measurements and then transmits its estimate to the fusion center. The problem of estimation fusion is to find an optimal state estimator based on all the received local estimates. Although the centralized fusion which directly makes use of all measurements from the local sensors in time is theoretically the best fusion strategy, sometimes communication or reliability constraints make it impossible to transmit all the sensor measurements to a fusion center. In contrast, the distributed fusion which only needs to fuse all received local estimates has the advantages of lower communication requirements, improved robustness, and so forth.

However, the fusion algorithms in distributed system have to deal with troubles that do not exist in centralized fusion. One of the difficulties is that the errors of local estimates to be fused are generally correlated, and as a result the distributed fusion cannot be achieved by a standard centralized algorithm such as the Kalman filter. The reasons of this correlation may be a common process noise in target when the state estimates are not fused at each sampling instant, or common prior information in the estimates from previous communication.

Over the last two decades, much research has been performed on distributed fusion [1–6]. Some approaches are looking for the "optimal" linear combination of local estimates in some criteria, such as weighted least squares or minimum variance [1, 2]. In [7], the authors proposed a new multi-sensor optimal information fusion criterion which is weighted by matrices in the linear minimum variance sense. An optimal Kalman filtering fusion with cross-correlated sensor noises is proposed in [8], which assumes that the correlation of sensor noises is accurately known. A unified model for estimation fusion based on the best linear unbiased estimation (BLUE) is proposed in [9]. However, all of the aforementioned methods rely on two assumptions: one is that the local estimates are unbiased and the other is that the error covariance matrix of all local estimates is known.

There are other approaches attempting to reconstruct the optimal centralized estimate from the local estimates. A random weighting estimation method for fusion of multidimensional position data is proposed in [10]. The method in [5, 6, 11] deduces to a linear combination of local estimates, but is not particularly effective in handling the correlation in measurement noises. In the seminal papers [4, 6, 12, 13], the covariance intersection (CI) algorithm was proposed to deal with this problem. It fuses without assuming any knowledge on the correlation between the local estimation errors. A robust estimation fusion is proposed in [14], which assumes that the correlation between the local estimation errors is not accurately known but belongs to an uncertain set. However, it is also a linear combination of local estimates as the other aforementioned methods. Theoretically, the linear combination may not be an accurate formation of the distributed fusion. Recently, a nonlinear estimation fusion is proposed in [15], where it minimizes the estimation error covariance only for the most favorable realizations of the random matrix and models it as an optimization problem with a chance constraint. Such optimization problem is also nonconvex and with appropriate relaxation it can be simplified to a convex problem. Similar with all the other aforementioned methods, it considers the optimal fusion in the sense of statistics, which do not necessarily lead to a small estimation error. There may be the case that the estimation error is very large even though the optimal criteria considered is small. So far, the robustness of the fusion estimation is still a challenge.

In this paper, we are looking forward to establishing a robust distributed fusion strategy under some basic assumptions. This robust fusion is aimed at minimizing the worst-case fusion error, which is achieved through a minimax problem. Although it is non-convex, we can relax it to a semidefinite program (SDP) following [16]. The resulted SDP problem can be solved quite efficiently in polynomial time by an interior point method; in particular, by the homogeneous self-dual method [17] or toolbox CVX in Matlab. Then the resulted fusion estimate is a form of a nonlinear combination of local estimates. Since the geometrical interpretation of our fusion method is in coincidence with the Chebyshev center, we call it the ion (CFE). The basic assumption of this paper is that the local estimation errors are bounded. Although it is not satisfied theoretically if the estimation error is a Gaussian distributed variable, it can be guaranteed in a nearly 100% probability if the bound is large enough and in practical applications it can always be satisfied. We call this bound the supporting bound, which is directly related to the resulted Chebyshev fusion estimate. So we further investigate the sensitive analysis of the relationship between the Chebyshev fusion estimate and the supporting bound. The result shows that the performance of the proposed Chebyshev fusion estimation is robust to the choice of the supporting bound. Moreover, numerical simulations are used to corroborate the theoretical results which demonstrate the good performance of the proposed CFE method.

The remainder is organized as follows. We briefly introduce the distributed estimation fusion problem in Section 2 and propose the robust CFE method in Section 3. The sensitive analysis about the choice of parameter R in CFE method is provided in Section 4, and some numerical simulations are carried out in Section 5. Section 6 gives conclusions.

2. Distributed Estimation Fusion Problem

Consider the following l-sensor distributed dynamic system:

$$\begin{aligned} \mathbf{x}_{t+1} &= \Phi \mathbf{x}_t + \mathbf{v}_t, \quad (t = 1, \ldots, T), \\ \mathbf{y}_t^i &= \mathbf{H}^i \mathbf{x}_t + \mathbf{w}_t^i, \quad (i = 1, \ldots, l), \end{aligned} \tag{1}$$

where $\mathbf{x}_t \in \mathbf{R}^n$ is the state vector, $\Phi \in \mathbf{R}^{n\times n}$ is the transition matrix, $\mathbf{y}_t^i \in \mathbf{R}^{m_i}$ and $\mathbf{H}^i \in \mathbf{R}^{m_i \times n}$, $i = 1, \ldots, l$, are the observations and measurement matrices of l local sensors respectively, and $\mathbf{v}_t \in \mathbf{R}^n$ and $\mathbf{w}_t^i \in \mathbf{R}^{m_i}$ are the process noise, and the measurement noise respectively, which are norm-bounded zero mean random processes with covariance matrices $E(\mathbf{v}_t \mathbf{v}_t') = \mathbf{V}, E(\mathbf{w}_t \mathbf{w}_t') = \mathbf{W}$ and independent across sensors and time t.

Kalman's filtering is the best known recursive least mean square (LMS) algorithm to optimally estimate the unknown state of a dynamic system for a single sensor. Thus, the unbiased estimates $\hat{\mathbf{x}}_t^i$ and corresponding error covariances $\mathbf{P}_t^i = E[(\mathbf{x}_t - \hat{\mathbf{x}}_t^i)(\mathbf{x}_t - \hat{\mathbf{x}}_t^i)^{'}]$ $(i = 1, \ldots, l)$ are available by the Kalman filter. The distributed fusion problem is to generate an "optimal" estimate $\hat{\mathbf{x}}_t$ from $\hat{\mathbf{x}}_t^i$ for $i = 1, \ldots, l$.

There are three possible architectures in distributed fusion depending on the sources of $\hat{\mathbf{x}}_t^i$ [6]. In this paper, we consider the "Arbitrary distributed fusion," that is, $\hat{\mathbf{x}}_t^i$ $(i = 1, \ldots, l)$ are l arbitrary estimates to be fused, and no prior information or memory is available. The main problem is caused by correlated estimation errors, because in general $\mathbf{P}_t^{ij} = E[(\mathbf{x}_t - \hat{\mathbf{x}}_t^i)(\mathbf{x}_t - \hat{\mathbf{x}}_t^j)'] \neq \mathbf{0}$ for $i \neq j$ and their values may not be known.

In order to simplify the derivations, we start by reformulating the local estimate $\hat{\mathbf{x}}_t^i$ in terms of a mixture of uncorrelated components $\mathbf{e}_t^i$. More specifically, let us define $\mathbf{e}_t^i \in \mathbf{R}^n$ to be the normalized random vector $\mathbf{e}_t^i = \mathbf{P}_t^{i\,-1/2}(\mathbf{x}_t - \hat{\mathbf{x}}_t^i)$ such that $E[\mathbf{e}_t^i] = \mathbf{0}$ and $E[\mathbf{e}_t^i \mathbf{e}_t^{i\,'}] = \mathbf{I}$. Moreover, because the noises of the dynamic system are norm bounded, we make the following assumption.

Assumption 1. There exists a ball of radius R_t that contains the entire support of the unknown distribution of $\mathbf{e}_t^i$ for all $i = 1, \ldots, l$. More specifically, there exists $R_t \geq 0$ such that

$$P\left(\left(\mathbf{x}_t - \hat{\mathbf{x}}_t^i\right)' \mathbf{P}_t^{i\,-1} \left(\mathbf{x}_t - \hat{\mathbf{x}}_t^i\right) \leq R_t^2\right) = 1. \tag{2}$$

We believe that Assumption 1 is reasonable, because in practice the estimation error of the local sensor is impossible to be infinitely large, and we can always find a bound on it. In practical applications, even when we have no additional information about $\mathbf{x}_t$ and $\mathbf{e}_t^i$, we believe that an educated and conservative guess about the magnitude of R_t is available. We will also revisit this issue in Section 4 where we discuss the sensitivity of the resulting fusion estimation with respect to

the choice of R_t. In the rest part of this paper, a robust fusion estimation strategy will be derived based on Assumption 1.

3. The Robust Chebyshev Fusion Estimation Strategy

3.1. The Minimax Fusion Strategy. The most widely used fusion strategy is calculating the "best" linear combination of local estimates to minimize some criteria in statistics, such as minimum variance or weighted least squares. However, there may be some nonlinear formations to fuse the local information that performs better, which is at least as good as the linear combination because the linear combination is a special case of non-linear formation.

Moreover, the optimal fusion strategy in statistical meaning is not necessarily to get a good estimate with respect to the estimation error $\|(\mathbf{x}_t - \hat{\mathbf{x}}_t)\|^2$. Especially for the methods which depend on the unknown correlated estimation errors $\mathbf{P}_t^{ij}$, the performance of the fusion result may be considerably poor when the estimated $\hat{\mathbf{P}}_t^{ij}$ are not accurate enough. Because of these uncertainties in the distributed fusion, we propose the following robust mini-max fusion estimation.

Based on Assumption 1, we have observed that the state $\mathbf{x}_t$ must lie in the ellipsoid $E_i = \{\mathbf{x} : (\mathbf{x} - \hat{\mathbf{x}}_t^i)' \mathbf{P}_t^{i^{-1}} (\mathbf{x} - \hat{\mathbf{x}}_t^i) \le R_t^2\}$, so the intersection of the l quadratic ellipsoids is nonempty, which is defined as

$$Q = \{\mathbf{x} : \ f_i(\mathbf{x}) = \mathbf{x}'\mathbf{A}_i\mathbf{x} + 2\mathbf{b}_i'\mathbf{x} + c_i \le 0, 1 \le i \le l\}, \quad (3)$$

where $\mathbf{A}_i = \mathbf{P}_t^{i^{-1}}$, $\mathbf{b}_i = -\mathbf{P}_t^{i^{-1}}\hat{\mathbf{x}}_t^i$, and $c_i = (\hat{\mathbf{x}}_t^i)'\mathbf{P}_t^{i^{-1}}\hat{\mathbf{x}}_t^i - R_t^2$. Therefore, we have $P(\mathbf{x}_t \in Q) = 1$. In order to get a robust fusion estimation without the information on correlated local estimation errors, we directly treat the estimation error and suggest minimizing the worst-case error over Q, which is equivalent to finding the Chebyshev center of Q:

$$\min_{\hat{\mathbf{x}} \in \mathbf{R}^n} \max_{\mathbf{x} \in Q} \|\hat{\mathbf{x}} - \mathbf{x}\|^2. \quad (4)$$

The geometrical interpretation of the Chebyshev center is the center of the minimum radius ball enclosing Q. Thus, problem (4) can be equivalently written as

$$\min_{\hat{\mathbf{x}}, r} \left\{ r : \|\hat{\mathbf{x}} - \mathbf{x}\|^2 \le r, \ \forall \mathbf{x} \in Q \right\}. \quad (5)$$

However, computing the Chebyshev center (4) is a difficult optimization problem in general, because the inner maximization is nonconvex quadratic problem. Recent research in the context of quadratic optimization [3] shows that the Chebyshev center can be calculated efficiently when Q is the intersection of two ellipsoids in the complex domain, despite the nonconvexity. While in the real domain and when there are more than two constraints, a relaxed Chebyshev center (RCC) is proposed in [16].

3.2. The Relaxed Chebyshev Center Fusion Estimation. The RCC of Q, which is denoted as $\hat{\mathbf{x}}_{\mathrm{RCC}}$, is obtained by replacing the non-convex inner maximization in (4) by its semidefinite relaxation and then solving the resulting convex-concave min-max problem, and for more details, one can refer to [16]. Therefore, an explicit representation of $\hat{\mathbf{x}}_{\mathrm{RCC}}$ can be achieved by the following theorem.

Theorem 2. *The RCC of Q is given by*

$$\hat{\mathbf{x}}_{RCC} = -\left(\sum_{i=1}^{l} \alpha_i \mathbf{A}_i\right)^{-1} \left(\sum_{i=1}^{l} \alpha_i \mathbf{b}_i\right), \quad (6)$$

where $\{\alpha_1, \alpha_2, \ldots, \alpha_l\}$ is an optimal solution of the following convex optimization problem in l variables:

$$\min_{\alpha_i} \quad \left\{ \left(\sum_{i=1}^{l} \alpha_i \mathbf{b}_i\right)' \left(\sum_{i=1}^{l} \alpha_i \mathbf{A}_i\right)^{-1} \left(\sum_{i=1}^{l} \alpha_i \mathbf{b}_i\right) - \sum_{i=1}^{l} \alpha_i c_i \right\} \quad (7)$$

$$\text{s.t.} \quad \sum_{i=1}^{l} \alpha_i \mathbf{A}_i^{-1} \succeq \mathbf{I}, \quad \alpha_i \ge 0, \ i = 1, 2, \ldots, l. \quad (8)$$

It is not difficult to cast the optimization problem (7) as the following SDP:

$$\min_{\alpha_i} \quad \left\{ t - \sum_{i=1}^{l} \alpha_i c_i \right\} \quad (9)$$

$$\text{s.t.} \begin{pmatrix} \sum_{i=1}^{l} \alpha_i \mathbf{A}_i & \sum_{i=1}^{l} \alpha_i \mathbf{b}_i \\ \sum_{i=1}^{l} \alpha_i \mathbf{b}_i' & t \end{pmatrix} \succeq 0 \quad (10)$$

$$\sum_{i=1}^{l} \alpha_i \mathbf{A}_i \succeq \mathbf{I}, \quad \alpha_i \ge 0, \ i = 1, 2, \ldots, l. \quad (11)$$

We see that the fusion estimate $\hat{\mathbf{x}}_{\mathrm{RCC}}$ is completely a nonlinear combination with all the available local information, including the estimates $\hat{\mathbf{x}}_t^i$ and error covariances $\mathbf{P}_t^i$, and the coefficients α_i are solved by an SDP (9), which can be calculated with high efficiency. The local estimates $\hat{\mathbf{x}}_t^i$ are just the fusion estimate $\hat{\mathbf{x}}_{\mathrm{CFE}}$ when $\alpha_j = \delta_{ij}$, where $\delta_{ij} = 1$ when $i = j$, and $\delta_{ij} = 0$ when $i \ne j$. From Proposition IV.2 in [16], $\hat{\mathbf{x}}_{\mathrm{CFE}}$ is unique and feasible. So the worst-case estimation error of $\hat{\mathbf{x}}_{\mathrm{CFE}}$ is smaller than or at least as small as that of local estimators in the relaxed sense.

Remark 3. Note that from the definition of Q given in (3) and Theorem 2, the optimal fusion coefficients α_i are actually relative to the local estimates $\hat{\mathbf{x}}_t^i$. Therefore, the optimal fusion coefficients α_i are time varying and need to be solved at every sampling time t. Fortunately, the optimization problem (9)–(11) is an SDP, which is a class of convex optimization problems and can be solved in polynomial time using efficient algorithms, such as the software package SeDuMi or CVX toolbox in MATLAB. Therefore, this could satisfy real-time processing when the number of sensors l is not too large.

Among the variables, $\mathbf{A}_i$ and $\mathbf{b}_i$, except c_i, are independent of R_t, that is, the bound of the support of $\mathbf{e}_t^i$. So in Section 4, we focus on the choice of R_t. In what follows, we shall drop the argument t without confusion for notational simplicity.

4. Choosing the Support Bound R

From the expression of $\hat{\mathbf{x}}_{\mathrm{RCC}}$ in (7), the fusion estimate is determined by the parameters α_i, which is the solution of the SDP problem (9). Because R appears only in the optimal object, the choice of R does not infect the feasible set of (9). First of all, we discuss the sensitivity of the choice of R in CFE of distributed fusion estimation.

4.1. The Sensitivity of the Choice of R. Let us write the SDP problem (9) in the standard literature on linear semidefinite programs by

$$\begin{aligned} (\mathrm{P}) \quad & \max \quad \mathbf{g}'\mathbf{y} \\ & \text{s.t.} \quad \mathcal{A}^*(\mathbf{y}) + \mathbf{S} = \mathbf{C} \quad \mathbf{S} \succeq \mathbf{0}, \end{aligned} \tag{12}$$

where $\mathcal{A}^*(\mathbf{y}) := \sum_{i=1}^{l+1} \mathbf{y}_i \mathbf{F}_i$, $\mathbf{g} = [c_1, \ldots, c_l, -1]'$, $\mathbf{y} = [\alpha_1, \ldots, \alpha_l, t]'$, for $i = 1, \ldots, l$, $\mathbf{E}_i = \mathrm{diag}(\mathbf{e}_i)$, $\mathbf{e}_i(j) = 1$ if $i = j$, else $\mathbf{e}_i(j) = 0$, and

$$\begin{aligned} \mathbf{F}_i &= \begin{pmatrix} \mathbf{A}_i & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_i & \mathbf{b}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{b}_i' & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{E}_i \end{pmatrix}_{(2n+l+1)\times(2n+l+1)}, \\ \mathbf{F}_{l+1} &= \begin{pmatrix} \mathbf{0}_{(2n\times 2n)} & 0 & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \end{pmatrix}_{(2n+l+1)\times(2n+l+1)}, \\ \mathbf{C} &= \begin{pmatrix} \mathbf{I}_{n\times n} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}_{(2n+l+1)\times(2n+l+1)}. \end{aligned} \tag{13}$$

The dual of the primal program is

$$\begin{aligned} (\mathrm{D}) \quad & \min \quad \mathbf{C} \bullet \mathbf{X} \\ & \text{s.t.} \quad \mathcal{A}(\mathbf{X}) = \mathbf{g} \qquad X \succeq \mathbf{0}, \end{aligned} \tag{14}$$

where $\mathbf{C} \bullet \mathbf{X} := \mathrm{trace}(\mathbf{C}'\mathbf{X})$ and $\mathcal{A}(\mathbf{X}) := [\mathbf{F}_1 \bullet \mathbf{X}, \ldots, \mathbf{F}_{l+1} \bullet \mathbf{X})]$. The discussion of the sensitivity of the choice of R is based on the following assumption.

Assumption 4. The programs (P) and (D) are strictly feasible and there exist $\overline{\mathbf{y}}$, $\overline{\mathbf{S}}$, and $\overline{\mathbf{X}}$ which are unique and strictly complementary solutions of (P) and (D), that is,

$$\begin{aligned} & \mathcal{A}(\overline{\mathbf{X}}) = \mathbf{g}, \quad \mathcal{A}^*(\overline{\mathbf{y}}) + \overline{\mathbf{S}} = \mathbf{C}, \quad \overline{\mathbf{X}}\,\overline{\mathbf{S}} = \mathbf{0}, \\ & \overline{\mathbf{S}} \succeq \mathbf{0}, \quad \overline{\mathbf{X}} \succeq \mathbf{0}, \quad \overline{\mathbf{X}} + \overline{\mathbf{S}} \succ \mathbf{0}. \end{aligned} \tag{15}$$

Based on the above assumption, we consider the solutions of the programs (P) and (D) when there is a perturbation $\delta\mathbf{g}$ on $\mathbf{g}$ with the following theorem.

Theorem 5. *If the programs (P) and (D) satisfy Assumption 4 and the data* $\mathbf{g}$ *is changed by sufficiently small perturbation* $\delta\mathbf{g}$*, then the optimal solutions of the perturbed semidefinite programs are differentiable functions of perturbation* $\delta\mathbf{g}$*. Moreover, the derivatives* $\dot{\mathbf{y}} := D\overline{\mathbf{y}}(\delta\mathbf{g})$, $\dot{\mathbf{S}} := D\overline{\mathbf{S}}(\delta\mathbf{g})$ *and* $\dot{\mathbf{X}} := D\overline{\mathbf{X}}(\delta\mathbf{g})$ *at* $\overline{\mathbf{y}}$, $\overline{\mathbf{S}}$, $\overline{\mathbf{X}}$ *satisfy*

$$\begin{aligned} \mathcal{A}^*(\dot{\mathbf{y}}) + \dot{\mathbf{S}} &= \mathbf{0}, \\ \mathcal{A}(\dot{\mathbf{X}}) &= \delta\mathbf{g}, \\ \dot{\mathbf{X}}\overline{\mathbf{S}} + \overline{\mathbf{X}}\dot{\mathbf{S}} &= \mathbf{0}. \end{aligned} \tag{16}$$

Remark 6. The perturbation $\delta\mathbf{g}$ does not infect the feasible set of (P), and so does Slater's condition of (P). By continuity, Slater's condition of (D) is also satisfied for all sufficiently small perturbation $\delta\mathbf{g}$. The result in this theorem is based on the fact that Assumption 4 is still satisfied when perturbed $\mathbf{g}$ by $\delta\mathbf{g}$.

Remark 7. The result in this theorem is a special case in Theorem 1 in [18], which gives a comprehensive sensitivity result on the perturbation of all data of programs (P) and (D). Thus, our theorem could be a direct corollary from it.

Remark 8. Although the derivatives $\dot{\mathbf{y}}$, $\dot{\mathbf{S}}$, and $\dot{\mathbf{X}}$ are characterized by a system of linear equations (16), it is an overdetermined system of $l + 1 + (2n + l + 1)(3n + 3l/2 + 2)$ linear equations for the $l+1+(2n+l+1)(2n+l+2)$ unknowns.

Theorem 9. *The derivatives* $\dot{\mathbf{y}}$, $\dot{\mathbf{S}}$, *and* $\dot{\mathbf{X}}$ *in* (16) *can be given as the unique solution of the following nonsingular system of* $l + 1 + (2n + l + 1)(2n + l + 2)$ *linear equations for the* $l + 1 + (2n + l + 1)(2n + l + 2)$ *unknowns.*

Proof. By the conditions in Assumption 4, $\overline{\mathbf{X}}\,\overline{\mathbf{S}} = \mathbf{0} = \overline{\mathbf{S}}\,\overline{\mathbf{X}}$, and thus the matrices $\overline{\mathbf{X}} \succeq \mathbf{0}$ and $\overline{\mathbf{S}} \succeq \mathbf{0}$ commute. This guarantees that there exists a unitary matrix $\mathbf{U}$ that simultaneously diagonalizes $\overline{\mathbf{S}}$ and $\overline{\mathbf{X}}$. Therefore, by Corollary 1 in [18], the derivatives $\dot{\mathbf{y}}$, $\dot{\mathbf{S}}$, and $\dot{\mathbf{X}}$ can be solved from the following system:

$$\begin{aligned} \mathcal{A}^*(\dot{\mathbf{y}}) + \dot{\mathbf{S}} &= \mathbf{0}, \\ \mathcal{A}(\dot{\mathbf{X}}) &= \delta\mathbf{g}, \\ \Pi_{\mathrm{up}}\left(\mathbf{U}'\left(\dot{\mathbf{X}}\overline{\mathbf{S}} + \overline{\mathbf{X}}\dot{\mathbf{S}}\right)\mathbf{U}\right) &= \mathbf{0}, \end{aligned} \tag{17}$$

where $\Pi_{\mathrm{up}}(\mathbf{X})$ denotes the upper triangular of $\mathbf{X}$. □

So far, we have theoretically analyzed the sensitivity of a perturbation $\delta\mathbf{g}$ for SDP (P). The derivatives of the optimal solution to the perturbation could be calculated by a nonsingular system of linear equations. Because the variable R only exists in the object parameter $\mathbf{g}$, the change of R leads to a perturbation $\delta\mathbf{g}$ on the direction $[1, \ldots, 1, 0]'$. If the value of $\dot{\mathbf{y}}$ is sufficiently small, the performance of the proposed CFE is robust due to the choice of R.

4.2. The Geometrical Interpretation of R. From the expression in (3), we see that R in fact determines the size of the l

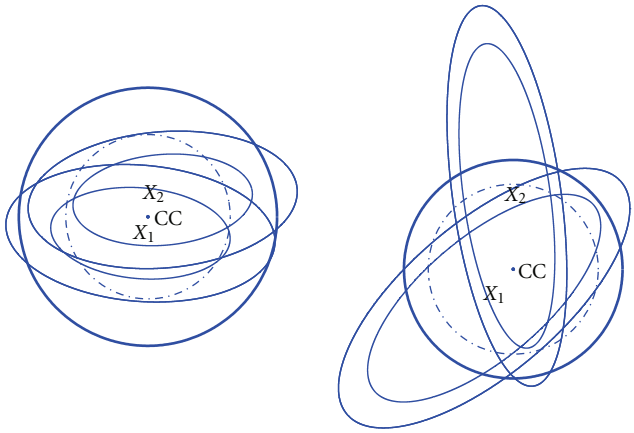


Figure 1: The illustration of the insensitivity on the choice of R.

ellipsoids. We illustrate in Figure 1, that the RCC of two interacting ellipsoids is still the same when changing the sizes simultaneously.

A geometrical interpretation about this phenomenon is that the RCC reflects the center point of the intersection of some ellipsoids in some sense. When simultaneously enlarges or reduces the sizes of these ellipsoids, the resulted RCC still represents the center location in the same sense, so it is not strange that the RCC is insensitive to the choice of R. In fact, as in the simulations in Section 5, we illustrate that the influence of the value of R on the fusion estimation is trivial.

However, we should certify that when changing the value of R, these ellipsoids own a common interaction area. Therefore, we suggest making a conservative choice of R. In practice, we can estimate it from the experienced learning or prior information.

Also from Figure 1, we see that the RCC of two ellipsoids may be either the linear combination of the centers of the ellipsoids or not. So the CFE varies a larger space comparing with the other linear fusion methods.

5. Simulation Experiments in Localization and Tracking

In this section, some simulation experiments are designed to show the performance of the proposed CFE method in localization and tracking and compare it with the result of the previous CI method. In addition, we have designed a numerical simulation to test the sensitiveness of the choice of the value R as well.

5.1. Simulation of Dynamic System. We consider the following dynamic system:

$$\begin{aligned} \mathbf{x}_{t+1} &= \Phi \mathbf{x}_t + \mathbf{v}_t, \quad (t = 1, \ldots, T) \\ \mathbf{y}_t^{(i)} &= \mathbf{H}^{(i)} \mathbf{x}_t + \mathbf{w}_t^{(i)} \quad (i = 1, 2). \end{aligned} \tag{18}$$

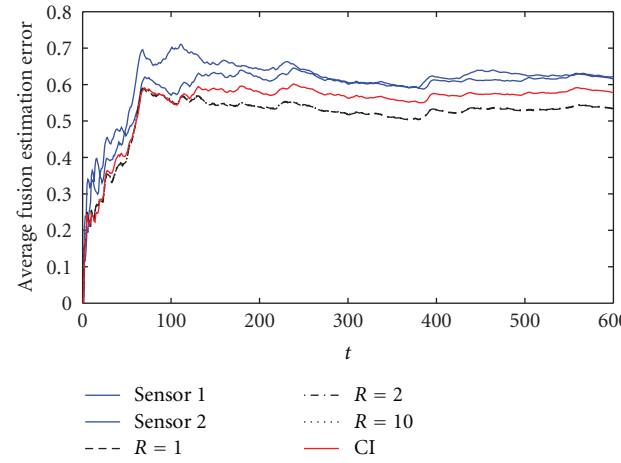


Figure 2: The average estimation error with respect to t for the local sensors, CFE, and CI method for $R_{\mathbf{v}} = [0.05\ 0; 0\ 0.05]$, $R_{\mathbf{w}}^{(i)} = [1\ 0; 0\ 2]$, where the CFE is calculated for $R = 1, 2, 10$ separately.

Case 1. Consider

$$\Phi = \begin{pmatrix} \cos\left(\frac{2\pi}{300}\right) & \sin\left(\frac{2\pi}{300}\right) \\ -\sin\left(\frac{2\pi}{300}\right) & \cos\left(\frac{2\pi}{300}\right) \end{pmatrix}, \quad \mathbf{H}_i = \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix}, \tag{19}$$

and the noises $\mathbf{v}_t$ and $\mathbf{w}_t^{(i)}$ are normally distributed with zero means and covariances $R_{\mathbf{v}} = [0.05\ 0; 0\ 0.05]$ and $R_{\mathbf{w}}^{(i)} = [1\ 0;\ 0\ 2]$, respectively. $\hat{\mathbf{x}}_t^i$ $(i = 1, 2)$ are 2 local estimators of $\mathbf{x}_t$ with covariance $\mathbf{P}_t^i$, respectively, which are calculated by a standard Kalman filter. The two sensors transmit their local estimates and covariance matrices to the fusion center, so it has the information of $\hat{\mathbf{x}}_t^i$ and $\mathbf{P}_t^i$.

We use the CFE and CI methods to fuse the two local estimates tracking the target for $t = 1, \ldots, 600$, where the CFE is calculated by solving the SDP problem (9) with the software package SeDuMi. The CI fusion is calculated following the method in [6]. The tracking performances are evaluated by the average estimation error, which is defined as

$$\mathrm{ARE}(\mathbf{x}_t) = \frac{\sum_{l=1}^{L} \left\| \overline{\mathbf{x}}_t^l - \mathbf{x}_t \right\|}{L}, \tag{20}$$

where $\overline{\mathbf{x}}_t^l$ denotes the estimation fusion of the state $\mathbf{x}_t$ at ensemble l and $L = 1000$ is the number of ensemble runs. The tracking performances of the local sensors, CFE, and CI method are illustrated in Figure 2, which shows the results of the average estimation error with respect to sampling time t for the local sensors, CFE, and CI method, respectively, where the CFE is calculated for $R = 1, 2, 10$ separately.

From Figure 2, we see that the average estimation error of CFE is consistently smaller than the local sensors as well as the CI method for all the choice of $R = 1, 2, 10$, which

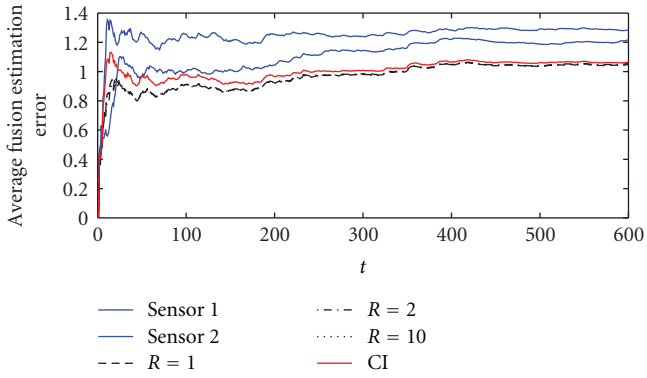


Figure 3: The average estimation error with respect to t for the local sensors, CFE, and CI method for $R_{\mathbf{v}} = [0.5\ \ 0; 0\ \ 0.5]$ and $R_{\mathbf{w}}^{(i)} = [3\ \ 0; 0\ \ 4]$, where the CFE is calculated for $R = 1, 2, 10$ separately.

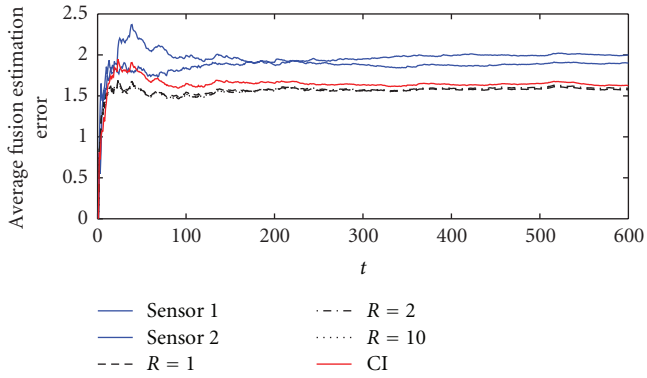


Figure 4: The average estimation error with respect to t for the local sensors, CFE, and CI method for $R_{\mathbf{v}} = [5\ \ 0; 0\ 5]$ and $R_{\mathbf{w}}^{(i)} = [3\ \ 0; 0\ \ 4]$, where the CFE is calculated for $R = 1, 2, 10$ separately.

verified that the proposed CFE method is more accurate compared with the CI method. Also, the average estimation errors are almost the same with respect to different values of R, which experimentally illustrate that CFE is insensitive to the value of R.

The next simulation is carried out for the same dynamic system as above, but the covariances of the noises $\mathbf{v}_t$ and $\mathbf{w}_t^{(i)}$ are $R_{\mathbf{v}} = [0.5\ \ 0; 0\ \ 0.5]$ and $R_{\mathbf{w}}^{(i)} = [3\ \ 0; 0\ \ 4]$, respectively. The resulted tracks and average estimation errors are shown in Figure 3. We can achieve the same results from this simulation that the CFE method is more accurate than CI method and the performances of CFE for different values of R are very close to each other.

Figure 4 is the tracks and average estimation errors when $R_{\mathbf{v}} = [5\ \ 0; 0\ \ 5]$ and $R_{\mathbf{w}}^{(i)} = [3\ \ 0; 0\ \ 4]$. The maximal estimation error through the process in the three simulations are listed in Table 1.

Case 2. Consider

$$\mathbf{H}_1 = \begin{pmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix}, \quad \mathbf{H}_2 = \begin{pmatrix} 1 & -0.25 \\ 0.25 & 1 \end{pmatrix}. \tag{21}$$

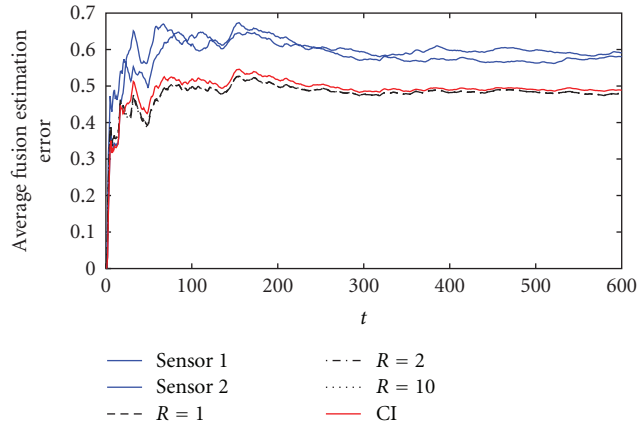


Figure 5: The average estimation error with respect to t for the local sensors, CFE, and CI method for $R_{\mathbf{v}} = [0.05\ \ 0; 0\ \ 0.05]$ and $R_{\mathbf{w}}^{(i)} = [1\ \ 0; 0\ \ 2]$, where the CFE is calculated for $R = 1, 2, 10$ separately.

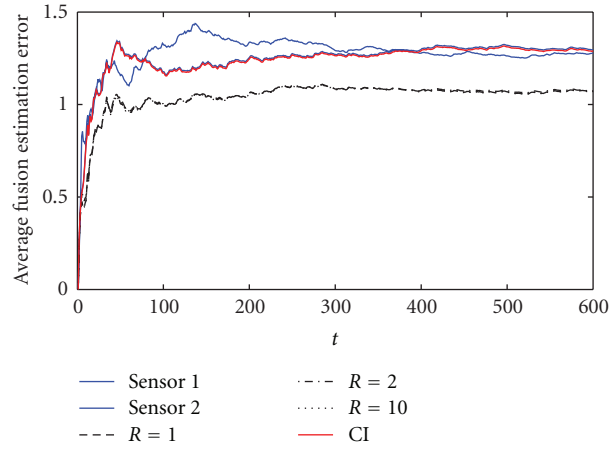


Figure 6: The average estimation error with respect to t for the local sensors, CFE, and CI method for $R_{\mathbf{v}} = [0.5\ \ 0; 0\ 0.5]$ and $R_{\mathbf{w}}^{(i)} = [3\ \ 0; 0\ \ 4]$, where the CFE is calculated for $R = 1, 2, 10$ separately.

In this case, also three simulations are carried out for different values of the covariances of the noises $\mathbf{v}_t$, and $R_{\mathbf{w}}^{(i)}$ respectively, and the other conditions are the same as in Case 1.

The tracks in this case are the same with Case 1. The average estimation errors through the process are illustrated in Figures 5–7. The improved performances of CFE are evidently better than CI when fusing the two local estimates, especially when the covariances of the noises are larger as in Figures 6 and 7. In fact, the performance of CI method in these two simulations are almost the same with local sensor 2, which is more accurate than local sensor 1. This comparison shows that CFE is a more stable method for distributed fusion because it always has a significant improvement when fusing the local estimates, while the CI method may just lead to the a local sensor estimate.

The maximal estimation error through the process in the three simulations are listed in Table 2.

Table 1: Comparison of the maximal estimation error through the process in the three simulations.

Simulation	Maximal squared estimation error					
	Sensor 1	Sensor 2	CFE $R = 1$	CFE $R = 2$	CFE $R = 10$	CI
1	3.3750	2.9707	2.7223	2.7601	2.7280	3.2137
2	14.0788	16.3843	9.0723	9.0220	9.1505	10.1641
3	43.6527	30.6748	20.7948	20.8013	21.3269	28.3822

Table 2: Comparison of the maximal estimation error through the process in the three simulations.

Simulation	Maximal squared estimation error					
	Sensor 1	Sensor 2	CFE $R = 1$	CFE $R = 2$	CFE $R = 10$	CI
1	3.2650	2.9283	2.1412	2.1670	2.1692	2.4373
2	15.3545	13.6833	11.8701	11.9099	11.9364	15.3535
3	38.7475	31.1737	27.2534	27.2287	27.3080	30.7442

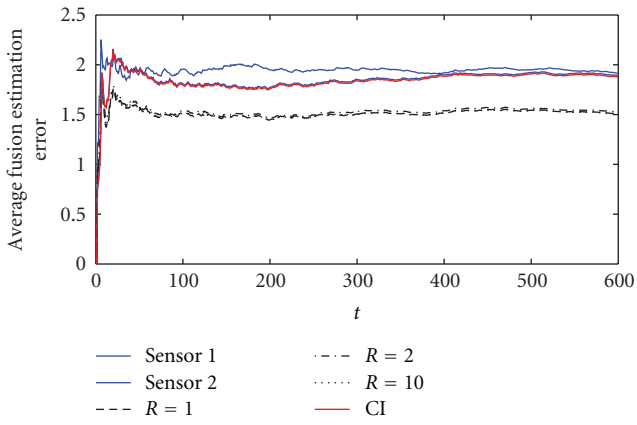


Figure 7: The average estimation error with respect to t for the local sensors, CFE, and CI method for $R_{\mathbf{v}} = [5\ \ 0; 0\ 5]$ and $R_{\mathbf{w}}^{(i)} = [3\ \ 0; 0\ \ 4]$, where the CFE is calculated for $R = 1, 2, 10$ separately.

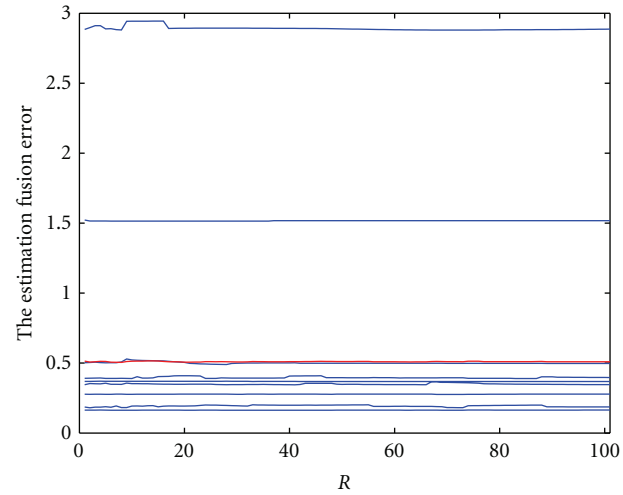


Figure 8: The estimation fusion error with respect to the value of R for 100 runs, where the blue line is the estimation error of the first 10 runs and the red line is the average estimation error for the 100 runs.

From Tables 1 and 2, we can see that the maximal estimation errors of CFE are much smaller than that of CI and the local sensors, which verified that the proposed CFE is a robust fusion estimation. Meanwhile, the performance of CFE is insensitive to the choice of R.

5.2. Sensitivity of the Value of R. In this simulation, we focus on the performance of CFE with respect to different values of R. This experiment explores the average estimation error by Monte-Carlos simulation. Suppose that the true initial state $\mathbf{x}_0$ and the local covariances of estimation error at this moment are known, that is, $\mathbf{x}_0 = [52.3246\ \ 2.2814]$, $\mathbf{P}_0^i = [0.2419\ \ -0.0456;\ \ -0.0456\ \ 0.2501]$ $(i = 1, 2)$. The dynamic system is the same as that of Case 1 in last the subsection and $R_{\mathbf{v}} = [1\ \ 0; 0\ \ 1]$ and $R_{\mathbf{w}}^{(i)} = [1\ \ 0; 0\ \ 2]$. We only consider the one step estimation fusion and use the CFE to fuse the one step estimates $\hat{\mathbf{x}}_1^1$ and $\hat{\mathbf{x}}_1^2$ when the value of R varies from 1 to 100.

The fused estimation error with respect to R for 100 runs illustrated in Figure 8, where the blue line is the estimation error of the first 10 runs and the red line is the average estimation error for the 100 runs with respect to R. From Figure 8, we see that the estimation error is nearly unchanged even when the value of R varies from 1 to 100, which verifies that the proposed CFE is not only a robust fusion but also a stable method for the choice of R.

6. Conclusions

In this paper, we propose a method using a mini-max strategy to get a robust fusion estimation in distributed multi-sensor systems for localization and tracking. This method is under the basic assumption that the normalized estimation error of local sensors are norm bounded, thus we can characterize the feasible set of the true state by the intersection of some ellipsoids. Then we proposed the mini-max fusion estimation in order to minimize the worst-case squared error. However, the resulted optimization problem is in fact looking for the Chebyshev center of the interaction of

the ellipsoids, which is non-convex in nature. We relax it and get an approximate Chebyshev center by solving a relaxed SDP problem. The resulted estimation fusion is not a linear combination of local estimates. Judging from the simulation results, the proposed CFE method is a robust estimation fusion and more accurate compared with the CI method.

Acknowledgments

This work was supported by the National Basic Research Program of China under Grant 2013CB329405 and the National Natural Science Foundation of China under Grants 61102007, 60974147, and 61032001.

References

[1] Y. Bar-Shalom, "On the track-to-track correlation problem," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 571–572, 1981.

[2] Y. Bar-Shalom and L. Campo, "The effect of the common process noise on the two-sensor fused track covariance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 22, pp. 803–805, 1986.

[3] A. Beck and Y. C. Eldar, "Regularization in regression with bounded noise: a chebyshev center approach," *SIAM Journal on Matrix Analysis and Applications*, vol. 29, no. 2, pp. 606–625, 2007.

[4] L. Chen, P. O. Arambel, and R. K. Mehra, "Estimation under unknown correlation: covariance intersection revisited," *IEEE Transactions on Automatic Control*, vol. 47, no. 11, pp. 1879–1882, 2002.

[5] C. Y. Chong, "Distributed fusion architectures and algorithms," in *Proceedings of the International Conference on Multisource-Multisensor Data Fusion*, 1998.

[6] C. Y. Chong and S. Mori, "Convex combination and covariance intersection algorithms in distributed fusion," in *Proceedings of the International Conference on Information Fusion (FUSION '01)*, August 2001.

[7] S.-L. Sun and Z.-L. Deng, "Multi-sensor optimal information fusion Kalman filter," *Automatica*, vol. 40, no. 6, pp. 1017–1023, 2004.

[8] E. Song, Y. Zhu, J. Zhou, and Z. You, "Optimal Kalman filtering fusion with cross-correlated sensor noises," *Automatica*, vol. 43, no. 8, pp. 1450–1456, 2007.

[9] X. R. Li, Y. Zhu, J. Wang, and C. Han, "Optimal linear estimation fusion—part I: unified fusion rules," *IEEE Transactions on Information Theory*, vol. 49, no. 9, pp. 2192–2208, 2003.

[10] S. Gao, Y. Zhong, and B. Shirinzadeh, "Random weighting estimation for fusion of multi-dimensional position data," *Information Sciences*, vol. 180, no. 24, pp. 4999–5007, 2010.

[11] R. Lobbia and M. Kent, "Data fusion of decentralized local tracker outputs," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 3, pp. 787–799, 1994.

[12] S. J. Julier and J. K. Uhlmann, "Non-divergent estimation algorithm in the presence of unknown correlations," in *Proceedings of the American Control Conference*, pp. 2369–2373, June 1997.

[13] S. Julier and J. Uhlmann, "General decentralized data fusion with covariance intersection (CI)," in *Handbook of Multisensor Data Fusion*, D. Hall and J. Llians, Eds., chapter 12, pp. 1–25, CRC Press, Boca Raton, Fla, USA, 2001.

[14] X. Qu, J. Zhou, E. Song, and Y. Zhu, "Minimax robust optimal estimation fusion in distributed multisensor systems with uncertainties," *IEEE Signal Processing Letters*, vol. 17, no. 9, pp. 811–814, 2010.

[15] D. Wu, J. Zhou, A. Hu, and F. Li, "Robust distributed fusion for system with randomly uncertain sensor estimation error cross-covariance," *International Journal of Systems Science*. In press.

[16] Y. C. Eldar, A. Beck, and M. Teboulle, "A minimax Chebyshev estimator for bounded error estimation," *IEEE Transactions on Signal Processing*, vol. 56, no. 4, pp. 1388–1397, 2008.

[17] J. F. Sturm, "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11-12, no. 1–4, pp. 625–653, 1999.

[18] R. W. Freund and F. Jarre, "A sensitivity result for semidefinite programs," *Operations Research Letters*, vol. 32, no. 2, pp. 126–132, 2004.