IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Particle Filtering-Based Indoor Positioning System for Beacon Tag Tracking

Yiwen Shen[1], (Student Member, IEEE), Beom Hwang[2], (Student Member, IEEE), and Jaehoon (Paul) Jeong.[2], (Member, IEEE)

[1]Department of Electrical & Computer Engineering, Sungkyunkwan University, Suwon, Republic of Korea (e-mail: chrisshen@skku.edu)
[2]Department of Computer Science & Engineering, Sungkyunkwan University, Suwon, Republic of Korea (e-mail: {hbum0147, pauljeong}@skku.edu)

Corresponding author: Jaehoon (Paul) Jeong (e-mail: pauljeong@skku.edu).

This work was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2017-0-01633) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation). This work was supported in part by IITP grant funded by the MSIT (No. 2019-0-01343, Regional strategic industry convergence security core talent training business). This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the MSIT (No. 2020R1F1A1048263).

ABSTRACT Indoor positioning technology is one of the cornerstones for many services in Internet-of-Things (IoT) systems. However, the existing indoor positioning systems are still suffering from several issues, such as unstable positioning accuracy, a high system complexity, and a high deployment cost. To tackle these issues, this paper presents a Particle Filter-based Indoor Positioning System (PFIPS) that can localize and track a tag that broadcasts Bluetooth Low Energy (BLE) beacon messages to BLE receivers. The proposed PFIPS uses a Kalman Filter to preprocess collected Received Signal Strength Indication (RSSI) information in order to smooth the fluctuated RSSI data. It also designs an effective Particle Filter (PF) to approximate the location of a tag, which gradually reduces the location uncertainties in a Gaussian belief space. To show the applicability of our PFIPS, we have developed PFIPS in a testbed based on commercial off-the-shelf (COTS) devices. Through intensive simulations and experiments, our experiment results show that our PFIPS outperforms the legacy indoor positioning systems in terms of location accuracy by 24.1% and achieves median accuracy of 1.16 m.

INDEX TERMS Indoor Positioning, Localization, Bluetooth Beacon, Particle Filter, Kalman Filter, RSSI.

I. INTRODUCTION

Internet-of-Things (IoT) systems can enable millions of devices to be connected. Positioning technology is one of the underlying technologies for IoT systems to be more intelligent [1]. For example, a central air-conditioning controller in a smart building by tracking pedestrian flows is able to intelligently adjust temperatures of different rooms in the building. In an indoor environment, sensing and positioning a target can rely on different tools, such as various proximity sensors and wireless signals. Proximity sensors (e.g., infrared sensors, motion sensors, and ultrasound sensors) help a positioning system sense a target, but it is difficult to provide the position information of the target [2]. Furthermore, placing many sensors indoors may significantly increase the deployment cost and reduce the maintainability. Localization systems based on vision sensors (e.g., cameras) that are powered by deep learning methods have shown extraordinary success [3], [4], yet the privacy issue has become a major concern for this kind of localization systems.

Recently, using wireless signals for indoor positioning has been advanced greatly. Wireless signals are ubiquitous especially in indoor environments, where all kinds of IoT devices connect to IoT hubs by different wireless communication technologies, e.g., WiFi, Bluetooth, and Zigbee. A wireless signal-based Indoor Positioning System (IPS) can employ a variety of information to localize a target, such as Received Signal Strength Indication (RSSI) [5], Channel State Information (CSI) [6], [7], Angle of Arrival (AoA) [8], Time of Flight (ToF), Time Difference of Arrival (TDoA) [9], and Phase of Arrival (PoA) [10]. Some of the information used for positioning requires particular devices or hardware setup. For instance, a CSI-based IPS needs to hack into a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3045610, IEEE Access

WiFi card to obtain CSI data [6], [7], and an AoA or TDoA-based IPS naturally involves an antenna array [8]. On the contrary, the RSSI information is easy to obtain from those wireless communication technologies, since many communication systems use the RSSI information to optimize their communication performance.

A. PROBLEM STATEMENT

Using RSSI information for indoor positioning can be an easy but sometimes unstable approach [5], since the RSSI information can be very noisy. The noises can come from the multi-path effect caused by signal reflections in an indoor environment, interference signals occupying the same or neighbor channels, and other factors. Because of this, simple trilateration approaches may fail to provide a satisfactory positioning service, which is different from an outdoor case that uses Global Positioning System (GPS). To improve accuracy and robustness of RSSI-based IPS, various approaches were proposed. For example, a dead reckoning approach [2] may combine RSSI and inertial information to localize and track a target, however, the system performance may still suffer from the noisy RSSI and inaccurate inertial data. Using fingerprinting of RSSI signals [11] can be another available approach but requires a site survey and a labor-intensive fingerprint collection process.

To use RSSI information in positioning a target, we can employ a probabilistic approach, such as a Bayesian estimation method and Particle Filter (PF) [12], which can generally provide better positioning accuracy. A PF uses the Bayesian theorem to approximate a location of the target. Through a history of RSSI information, a belief of the location of a target can be narrowed down to a small area gradually. The authors in [13] presented a PF-based reinforcement learning approach to conduct an indoor localization task by fusing several kinds of information, such as a radio communication range, Inertial Measurement Units (IMU), and floor plan (i.e., the description of a floor layout). However, in different scenarios, implementations of a PF for indoor localization can be quite different, and the detailed PF process can also vary case by case.

B. MOTIVATION

The RSSI information can be extracted from different sources, e.g., WiFi and Bluetooth. Recently, the Bluetooth Low Energy (BLE) [14] technology gains considerable attentions since it is power efficient and easy to access via commercial off-the-shelf (COTS) devices. A BLE device can periodically broadcast beacons for months and years with only a coin cell battery, which is an ideal solution in many scenarios, such as remote sensing, remote health care, and smart building management. The latest Bluetooth 5.x version improves the BLE data rate up to 2 Mbps from the previous 1 Mbps and the coverage range up to 500 m from the previous 100 m [14]. A recent advancement in BLE research brings CSI to BLE-based indoor localization area [15], which was not possible previously due to a different PHY layer scheme in comparison with WiFi. However, designing an applicable IPS based on RSSI using COTS devices is still a challenging task.

C. CONTRIBUTIONS

Therefore, to further improve the localization accuracy and explore an applicable architecture with COTS devices, this paper proposes an effective Particle Filter-based IPS (PFIPS) for tracking tags that broadcast BLE beacons. The proposed PFIPS uses a Kalman Filter (KF) to preprocess RSSI information and a PF algorithm to approximate the location of a tag in an indoor environment. The tag periodically broadcasts beacon messages that include its ID and MAC address. An Anchor Point (AP) deployed at corners and boundary areas of a room can receive the information in the beacon messages from the tag and forward it to a server along with its ID and location information. The server estimates a location of the tag in the room by collecting all the information from several APs. We conducted both simulations and experiments to demonstrate the performance of our PFIPS, and our experiment results show that our PFIPS improves the localization error by 24.1% compared with a state-of-the-art scheme [16]. Our testbed uses a set of COTS devices, which shows that our PFIPS can be easily deployed in a particular indoor environment. Note that the main idea for the proposed PFIPS in this paper is demonstrated in our previous work [16], [17], and this paper improves the previous work in terms of system design, localization workflow, and target motion modeling for better localization accuracy. Our main contributions are as follows.

- A system design for the particle filter-based indoor positioning system that can be applied to different indoor positioning scenarios.
- A Gaussian model-based state update process and an RSSI gradient-based motion estimate model for the proposed PFIPS.
- The Proof of Concept (PoC) through the simulation of our PFIPS with a virtual indoor environment and the experiment of it with COTS devices that demonstrates the applicability of our localization system.

The rest of this paper is organized as follows. Section II summarizes the related work. Section III describes the problems we consider in this paper. Section IV introduces the background knowledge for the proposed system. Section V presents the design of our PFIPS. Section VI shows the performance evaluation by simulations and experiments. Section VII enumerates several use cases that can be built on our PFIPS. Section VIII gives a few limitations of our PFIPS. Finally, in Section IX, we conclude this paper along with future work.

II. RELATED WORK

Researchers have explored different kinds of approaches to position targets at an indoor environment. The RSSI information is widely used for indoor localization. Approaches using RSSI information can be categorized into three classes:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3045610, IEEE Access

TABLE 1: Comparison of Different Indoor Positioning Schemes

Name	Technology	Technique	Accuracy	Pros	Cons	Year	Ref.
PLS	CSI	Fingerprinting & Machine learning	0.5∼1.3 m	Positioning Non-Line-of-Sight (NLoS) targets	Complex CSI preprocessing step	2020	[18]
AngLoc	CSI & RSSI	AoA & ToF & Fingerprinting	0.3∼1.2 m	Joint angle and delay estimation	Offline angle data construction	2020	[8]
SP	RSSI	Moving Average & Particle Filter	1∼3 m	System simplicity	Distance estimation in one-dimension	2020	[19]
EasiTrack	CSI & IMU	Multilateration & Particle Filter	0.1∼1 m	High positioning accuracy; Single WiFi device with MIMO	CSI phase information loss; Requiring a high computation power for signal processing	2020	[20]
DLoc	CSI	AoA & ToF & Deep learning	0.4∼1 m	A deep neural network framework for CSI-based localization; Open CSI heatmap data set	Labor-intensive work for building and collecting CSI heatmaps	2020	[21]
STUPEFY	RSSI	Machine learning & Particle Filter	0.3∼1.2 m	High positioning accuracy	Offline training process	2019	[22]
SALA	RSSI & IMU	Centroid & PDR	2∼5 m	IoT devices assisted localization	Accuracy deviation of IMU	2018	[23]
Tril	RSSI	Multilateration	3∼10 m	System simplicity	Low positioning accuracy; Volatile system performance	2017	[24]

i) distance-based positioning techniques; ii) fingerprinting-based techniques; iii) probability estimation techniques.

A distance-based position technique can use triangulation or multilateration methods to localize a target. A Smartphone Assisted Localization Algorithm (SALA) [23] was proposed to localize IoT devices. It uses a distance-based centroid approach to localize both the smartphone itself and the IoT devices with a help of an IMU of a smartphone and RSSI information of beacon messages broadcasted by IoT devices. However, it suffers from the inaccurate readings of IMU, causing less accurate localization performance.

Fingerprinting methods [11] usually build a signal strength database to infer a possible location of a target via a training model. This type of approaches may require a person or a robot to collect the signal strength fingerprints of APs deployed inside a building, which can bring labor-intensive work in making the localization system work. AP-Sequence [25] is a fingerprint-based indoor localization system that requires low-overhead of fingerprint data building and maintenance. It simplifies the signal strength database construction by assigning unique AP-sequences to multiple partitioned region of an indoor area.

Some other approaches [26], [27] combine multiple techniques in positioning targets. The authors in [26] proposed a smartphone built-in sensor based Pedestrian Dead Reckoning (PDR) localization scheme by an enhanced particle filter and the WiFi Fine Timing Measurement (FTM) technique. The authors in [27] introduced an indoor smartphone localization approach that employs the round trip time of WiFi FTM and RSSI information.

STUPEFY [22] is a set-valued box PF-based BLE indoor localization approach. It uses the Kalman filtering technique to smooth RSSI data and constructs a Gaussian model for a fingerprinting approach. To determine a coarse box area where a target can be, it uses a learning-based coarse estimation method. After finding a coarse box area, STUPEFY operates a PF only in the identified area. Although STUPEFY achieves a high localization accuracy, it needs an offline training process for its Gaussian model in the coarse estimation.

The authors in [19] designed a BLE RSSI-based smart parking system by a PF method (called SP). The primary purpose of the SP system is to guide vehicle users to available parking lots as well as automatically charge parking fee when a vehicle leaves a parking lot. For checking parking lots availability, the SP system developed a PF algorithm to measure distances between a vehicle and surrounding parking lots. In the PF algorithm, they used a moving-average approach to smooth received BLE RSSI data. Since only the distances need to be estimated, the PF algorithm employs the univariate Gaussian distribution to calculate and update weights of the particles.

Recently, CSI-based indoor positioning approaches are gaining momentum in the community, since CSI in wireless signals can provide more information about an environment including activities of humans [28]. To deal with practical issues in a CSI-based indoor localization system, the authors in [18] proposed a view-selective deep learning approach that efficiently recovers the lost CSI elements and removes the non-informative CSI data with a non-negative matrix factorization method. But the complex preprocessing step implies the need of more computation power. AngLoc [8] is a CSI entropy-based AoA-aware probabilistic indoor localization system that uses a bivariate kernel regression scheme combining both amplitude and angular signatures. In the preprocessing step, AngLoc uses a power-based tap-filtering scheme and several phase calibration techniques to denoise CSI components. Like other fingerprinting schemes, AngLoc also needs to collect offline angle data for its database construction. The authors in [21] presented a deep-learning based indoor navigation framework (called DLoc) by an environment mapping approach. DLoc uses a CSI-based AoA and ToF combined method to localize a target. However, different from traditional methods, DLoc builds many 2D AoA and ToF heatmaps translated from CSI to be used in its deep-learning algorithm for localizing targets. Nevertheless, construction of the heatmaps may become a labor-intensive

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3045610, IEEE Access

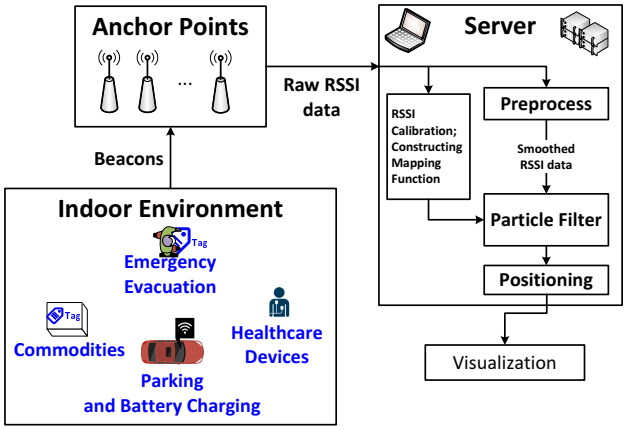


FIGURE 1: The system architecture of the proposed IPS.

work. EasiTrack [20] is an indoor tracking system that can reach a decimeter-level accuracy by using multipath CSI with virtual antennas and orientation from an IMU. It integrates the two information sources into a map-augmented particle filtering process to locate and track a target. Similar to other CSI-based approaches, the CSI phase information can be lost sometimes, which may reduce its applicability in a real system.

Table 1 shows a comparison among different indoor positioning schemes, which includes scheme name, technology used, localization technique, accuracy, pros and cons as well as publication year. Different from the existing approaches, this work focuses on both a system-level design and the PF-based indoor positioning procedure using the BLE RSSI information for moving tags. We propose a system architecture for the indoor positioning that can be applied to different indoor scenarios. The process of the PF-based indoor positioning in our system adopts a Gaussian model for state update and designs an RSSI gradient-based motion estimate model. In the next section, we will illustrate the problem formulation of our PFIPS.

III. PROBLEM FORMULATION

This section formulates the problem for an indoor localization system. We introduce the architecture of the proposed system; next we highlight the overall system and assumptions.

A. SYSTEM ARCHITECTURE

In this paper, we propose an indoor localization system as shown in Fig. 1. In an indoor environment, a tag running by a button cell battery or a wearable device using a rechargeable battery can broadcast beacon messages periodically. Multiple beacon receivers (i.e., APs), which are deployed at corners and boundaries of the room with known locations (i.e., coordinates), can receive these beacon messages. By upon receiving a beacon message, an AP can obtain the information contained in the message and the RSS information from its physical layer as well. This information is forwarded toward a server. The server preprocesses the raw RSSI data and inputs the preprocessed data into a particle filter for localization. Through recursively calculations, we can approximate the location of a tag. This kind of architecture can be applied for the use cases mentioned in Section VII.

B. OVERVIEW

In detail, as shown in Fig. 1, the proposed IPS works as follows.

- A server running the proposed IPS collects and preprocesses raw RSSI data by a KF.
- Meanwhile, the server also needs to construct an RSSI-to-distance mapping function, which will be used for PF procedure.
- The preprocess step for the received raw RSSI data by a KF is to smooth the data to reduce fluctuations, which employs a general update and prediction procedure.
- The smoothed RSSI data are put into a PF that has several steps, such as initialization, state update, resampling, and location estimation.
- *Initialization*: It virtually generates a set of particles with random locations in an indoor environment. A weight matrix derived by the room layout can also be collected to reduce less likely locations caused by different indoor facilities.
- *State Update*: This step updates the state of particles, which uses a set of latest filtered RSSI data by an empirical mapping function between an RSSI and a distance.
- *Effective Particle Checking*: It evaluates the quality of all particles in the system to determine whether to do the next resampling step.
- *Resampling and Weight Normalization*: It updates all the generated particles with replacement to remove some of them with low likelihood. Every time step it may run once or not depending the effective particle volume, which is described in detail in Section V-C4.
- *Location Estimation*: It estimates the location of a target by the sampled particles.

While beacon transmitters keep broadcasting beacons, the server can continuously localize the transmitters. A transmitted beacon message contains the transmitter's ID and MAC address information. As shown in Fig. 1, the beacon transmitters can be attached to commodities for logistics, mounted on vehicles for a smart transportation, and embedded into a wearable device for healthcare.

C. ASSUMPTIONS

We have the following assumptions:

- The deployment locations of APs (i.e., coordinates of beacon receivers) are assumed to be known.
- The Line-of-Sight (LoS) and Non-LoS (NLoS) targets for positioning are not separately considered, since the scenarios that we are considering have both cases.
- Both 1-dimension (1D) and 2-dimension (2D) scenarios are considered, and 3-dimension (3D) cases are left as future work.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3045610, IEEE Access

In the next section, we will introduce some preliminary knowledge for the proposed IPS.

IV. PRELIMINARY

Our system uses a KF to preprocess received RSSI data and a particle filter to localize targets. In this section, we briefly introduce the Kalman and particle filtering process.

A. KALMAN FILTERING

Kalman filtering techniques have been considered as the optimal solution for many signal processing and prediction tasks that have a linear system model with Gaussian noise [29]. A KF generally employs a *prediction* and *update* procedure to obtain states of a system. Assuming that a system can be described by

$$s_{t+1} = \Phi s_t + u_t, \tag{1}$$

where s_t is the state vector of the system at the time step t; Φ is the state transition matrix of the system from time step t to $t+1$; u_t is a known white noise process that has a known covariance matrix Q. We also have a measurement equation for the system as

$$m_t = H s_t + v_t, \tag{2}$$

where m_t is a measurement state vector of s at time step t; H is a connection matrix for the measurement; v_t is a measurement error, which is considered as a known white noise process with a known covariance matrix R. The error covariances Q of the system and R of the measurement can be expressed by $Q = E[u_t u_t^T]$ and $R = E[v_t v_t^T]$, respectively.

In a KF, a state update equation can be expressed by

$$\hat{s}_t = \hat{s}'_t + K_t(m_t - H s'_t), \tag{3}$$

where $\hat{s}_t$ is an estimate of the system state s_t, and the prior estimate of $\hat{s}_t$ is denoted by $\hat{s}'_t$; K_t is the *Kalman gain* that can be obtained by

$$K_t = P'_t H^T (H P'_t H^T + R)^{-1}. \tag{4}$$

The mean square error (MSE) covariance P_t of the system can be given by

$$P_t = (I - K_t H) P'_t, \tag{5}$$

where I is an identity matrix, and the prior MSE covariance P'_t is updated by

$$P'_{t+1} = \Phi P_t \Phi^T + Q, \tag{6}$$

and the prior system state for next time step is given by

$$\hat{s}'_{t+1} = \Phi \hat{s}_t. \tag{7}$$

Through the above prediction and update procedure, we can filter noises to obtain the true system states.

However, for a nonlinear system, a KF may not be able to provide a desirable estimate for the system states. For such a kind of systems, usually approximation approaches are considered as a better choice, such as Bayesian estimation and particle filtering.

B. PARTICLE FILTERING

A *Bayesian estimation* method may provide a better estimation accuracy in a nonlinear system. Similar to the system description of the KF in (1), a nonlinear system for its state s^t at time step t can be expressed in another form as

$$s^{t+1} = f^t(s^t, u^t), \tag{8}$$

and a history of measurement values M^t (i.e., m^1, m^2, ..., m^t) is given by

$$m^t = h^t(s^t, v^t), \tag{9}$$

where $f^t(\cdot)$ and $h^t(\cdot)$ are time-varying nonlinear system and measurement equations, respectively; u^t and v^t are independent noise processes. We want to approximate a *posterior* conditional probability density function (PDF) of s^t by the Bayesian rule [12] as

$$p(s^t|M^t) = \frac{p(m^t|s^t)p(s^t|M^{t-1})}{p(m^t|M^{t-1})}, \tag{10}$$

and $p(m^t|M^{t-1})$ can be calculated by the total probability law as

$$p(m^t|M^{t-1}) = \int p(m^t|s^t)p(s^t|M^{t-1})dm^t, \tag{11}$$

where $p(s^t|M^{t-1})$ is a *prior* probability of s^t that can be calculated by the Chapman–Kolmogorov equation [12] as

$$p(s^t|M^{t-1}) = \int p(s^t|s^{t-1})p(s^{t-1}|M^{t-1})ds^{t-1}, \tag{12}$$

in which $p(s^{t-1}|M^{t-1})$ is known at the beginning time of the system as

$$p(s^0|M^0) = p(s^0), \tag{13}$$

and $p(s^t|s^{t-1})$ is known by the system equation (8).

The above Bayesian estimation process estimates the current state s^t of (10) by an *update* of (11) and a *prediction* of (12), however, it cannot be directly applied to a nonlinear system since the posterior PDF of s^t cannot be calculated analytically [12]. Therefore, a *particle filtering* process was proposed to use this process to recursively estimate the state of a target [19] in a discrete way. We can randomly generate a set of N state vectors by the initial distribution of s, which is assumed to be known. Each element of the set N is called a particle denoted by $P_i^t = \{s_i^t, w_i^t\}, i \in \{1, ..., |N|\}$ at time t, where w_i^t is a weight (i.e., probability) for this particle. The state of a particle P_i^t can be updated at each time step [29] by the equation (8). We can calculate the relative likelihood w_i^t of a particle P_i^t by evaluating the PDF $p(m^t|s_i^t)$ with the nonlinear measurement function (9), which can be expressed by

$$w_i^t = w_i^{t-1} \cdot \frac{p(m^t|s_i^t)p(s_i^t|s_i^{t-1})}{q(s_i^t|s_i^{t-1}, m^t)}, \tag{14}$$

where $q(\cdot)$ is a proposal of the important density that reflects

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3045610, IEEE Access

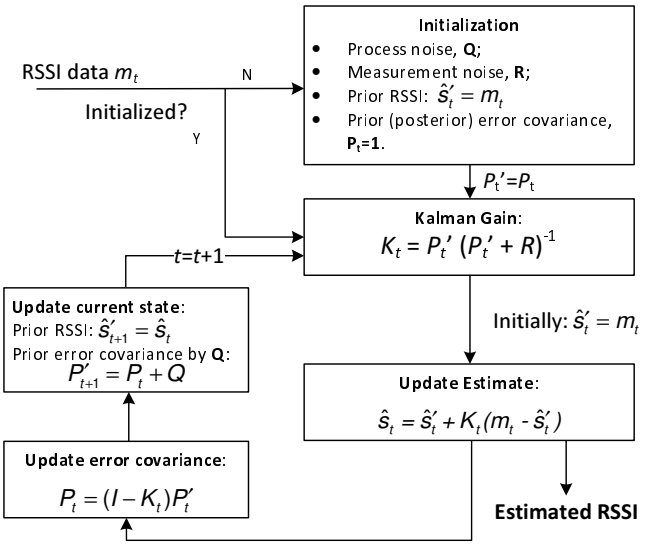


FIGURE 2: RSSI data smoothing module by a KF.

the state of the current system.

Then, the posterior density of s^t at time step t can be approximated by the Dirac delta function $\delta(\cdot)$ [12] as

$$p(s^t|M^t) \approx \sum_{i=1}^{|N|} w_i^t \delta(s^t - s_i^t), \tag{15}$$

where w_i^t is a normalized weight of a particle P_i^t such that $\sum_i^{|N|} w_i^t = 1$, and s^t is the system state and s_i^t is a particle P_i^t state at time step t. At the next time step $t+1$, we can either update the weight w_i^t or resample the generated particles with replacement such that the new sampled particles follow the distribution of w_i^t.

In practice, a particle filtering process may suffer from the *degeneracy problem* and the *sample impoverishment problem* [29]. Some remedy approaches are proposed to mitigate these problems, such as using an *effective sample size* for the degeneracy problem [19] and *Markov Chain Monte Carlo* resampling [30], [31] for the impoverishment problem. In the next section, we will introduce the design of the proposed PFIPS.

V. PARTICLE FILTER-BASED INDOOR POSITIONING SYSTEM

In this section, we present the design of our IPS. We first describe how we smooth raw RSSI data by a KF; next, we refer to an RSS-to-distance mapping model; eventually, we show the design of localization using a particle filter for our PFIPS.

A. RSSI SMOOTHING BY KALMAN FILTER

Based on the architecture in Fig. 1, when raw RSSI data are collected by APs, due to the environment noises, the raw RSSI data may fluctuate severely, which may cause a large variance for estimation. To reduce this kind of drawback, we propose a KF to smooth the raw RSSI data.

Fig. 2 shows a diagram of RSSI data smoothing process by a KF. The original RSSI data m_t uploaded by the APs at time step t are input. At the beginning, a prior RSSI data point $\hat{s}'_t$ is set to equal m_t, and a prior error covariance P_t equals 1, where a posterior error covariance P'_t equals P_t. Here, since the RSSI data are one-dimensional, the error covariance matrix in a standard KF is reduced to a scalar. We assume that a process noise Q and a measurement noise R are known. Thus, we can compute a Kalman gain K_t by

$$K_t = P'_t(P'_t + R)^{-1}, \tag{16}$$

and estimate a posterior RSSI value $\hat{s}_t$ by

$$\hat{s}_t = \hat{s}'_t + K_t(m_t - s'_t). \tag{17}$$

This posterior RSSI value $\hat{s}_t$ is a predicted and smoothed RSSI data point for output, as shown in Fig. 2. Meanwhile, we also need to update the current error covariance by

$$P_t = (I - K_t)P'_t, \tag{18}$$

where I is an identity matrix with one dimension (i.e., a scalar with value 1). Next, we need to update the current state with the estimate and the error covariance by

$$P'_{t+1} = P_t + Q, \tag{19}$$

$$\hat{s}'_{t+1} = \hat{s}_t. \tag{20}$$

At the next time step, the posterior state $\hat{s}_t$ and error covariance P'_t become the prior information for the prediction. In practice, the selection of the process noise Q and measurement noise R can be adjusted according to an environment. By this Kalman filtering process, jitters of RSSI data can be reduced. With the smoothed RSSI data, we also need to map them to distances, which is described in the next section.

B. RSSI MEASUREMENT AND MAPPING

An RSSI value can be mapped to a distance d, and it is usually modeled [5] as

$$RSSI = -10n \log_{10} d + a, \tag{21}$$

where n is a path loss exponent and a is a base RSSI value measured by a receiver at a certain distance from a beacon transmitter. From a practical point of view, this mapping function needs to be evaluated by a specific experiment device used for our PFIPS. Based on (21), given an RSSI value, we can obtain a distance by

$$d = 10^{\frac{a - RSSI}{10n}}. \tag{22}$$

For the parameters n and a, we obtain them by experiments described in Section VI-B.

C. LOCALIZATION WITH PARTICLE FILTER

Although outliers of RSSI data can be largely removed by the proposed smoothing process in Section IV-A, a naive trilateration approach by the smoothed RSSI values may still not be able to provide an acceptable localization accuracy.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3045610, IEEE Access

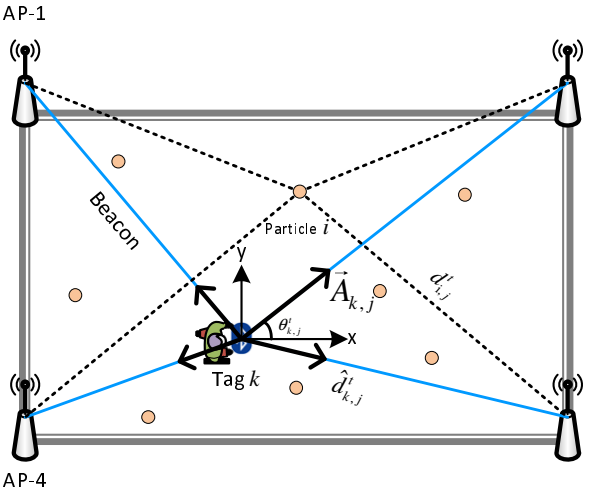


FIGURE 3: A scenario of indoor localization with BLE beacons and particle filter.

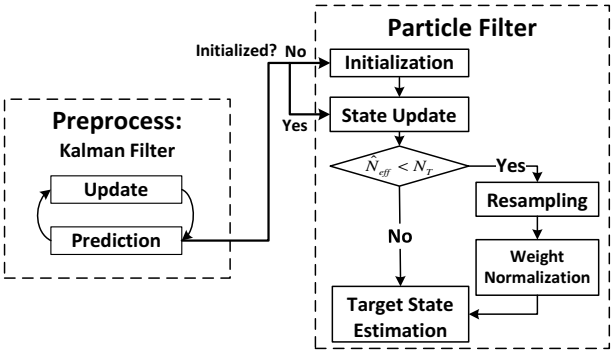


FIGURE 4: The workflow of the proposed IPS.

Therefore, in this section, we propose a particle filter approach to position a target for our IPS so that a better localization accuracy can be reached. A particle filter generally can include several steps as follows: i) initialization; ii) particle state and weight update; iii) resampling; and iv) target state estimation, as shown in Fig. 4.

1) State Definition and Initialization

Without loss of generality, we define a state of tag k at time step t as

$$s^t_k = \{(x^t_k, y^t_k), v^t_k, \alpha^t_k\}, \tag{23}$$

where (x^t_k, y^t_k), v^t_k, and α^t_k are the coordinates, speed, and heading angle of tag k, respectively. The initial speed v^0_k is assumed to be zero. A heading angle α^t_k is defined as the angle measured counterclockwise from x-axis to a heading direction of a tag. While a person carrying a tag moves around in a room, the state of the tag changes accordingly.

In the considered scenario, the only sensing information that we can obtain for tracking a tag is the received RSSI data in each AP. Thus, we model the heading angle α_k by using gradients of the RSSI data that describe the changing rate of it. If a gradient of RSSI data received by a AP is positive, it means that the tag is moving toward this AP, and vice versa. Although the RSSI data are filtered and smoothed by the smoothing step, gradients derived from them are still noisy. Therefore, we employ a digital signal filter to remove those high-frequency parts of the gradients data, since the gradients have a certain relationship with a moving speed of the tag, which in general has a lower frequency. Formally, as shown in Fig. 3, we can have an angle $\theta^t_{k,j}$ for a vector $\overrightarrow{A_{k,j}}$ from the current estimated position of a tag k to an AP j by

$$\theta^t_{k,j} = \arccos \frac{x_j - \hat{x}^t_k}{\hat{d}^t_{k,j}}, \tag{24}$$

where x_j and $\hat{x}^t_k$ are the x-coordinates of the AP j and the tag k, respectively. The current estimated position can have $|J|$ vectors having corresponding angles $\theta^t_{k,j}$ and magnitudes $c^t_{k,j}$ (i.e., gradients). By combining these $|J|$ vectors into one vector, we can determine an estimated heading direction for the tag, which can be expressed by

$$\overrightarrow{\hat{H}_k} = \sum_{j=1}^{J} \overrightarrow{A_{k,j}}. \tag{25}$$

An estimated speed $\hat{v}^t_k$ of the tag k can be derived from the magnitude $c^t_{k,j}$ by

$$\hat{v}^t_k = \beta \cdot c^t_{k,j}, \tag{26}$$

where β is a coefficient that scales the magnitude to a general speed measure, and in our system, it is set to 4.

A particle i has the same format of state $s^t_i = \{(x^t_i, y^t_i), v^t_i, \alpha^t_i\}$ to the tag k at time step t. Initially, N particles are generated and distributed in a room uniformly with a velocity $v^0_i = 0$ and a random heading angle α^0_i. Based on a practical RSSI mapping function mentioned in Section V-B, an estimated distance $\hat{d}^t_{k,j}$ between a tag k and an AP j can be obtained by the preprocessed RSSI data at time t, as shown in Fig. 3. Since the location of a particle $i \in \{1, ..., N\}$ is known, we can accurately calculate the distance between the particle i and an AP j as $d^t_{i,j}$. In the next step, we need to predict states of particles by the received RSSI information for approximating the state of a tag.

2) State Update

The predicted state of a particle i at time step t is represented by

$$\hat{s}^t_i = \{(\hat{x}^t_i, \hat{y}^t_i), \hat{v}^t_i, \hat{\alpha}^t_i\}. \tag{27}$$

A predicated position $(\hat{x}^t_i, \hat{y}^t_i)$ of a particle i [32] can be expressed by

$$\begin{aligned} \hat{x}^t_i &= x^{t-1}_i + v^{t-1}_i \Delta t \cdot \cos \alpha^{t-1}_i + \epsilon^t_x, \\ \hat{y}^t_i &= y^{t-1}_i + v^{t-1}_i \Delta t \cdot \sin \alpha^{t-1}_i + \epsilon^t_y, \end{aligned} \tag{28}$$

where ϵ^t_x and ϵ^t_y are Gaussian noise at time step t and Δt is the actual time interval between time step $t-1$ and t. The predicted speed $\hat{v}^t_i$ is assumed to be equal to the estimated speed $\hat{v}^t_k$ of tag k by

$$\hat{v}^t_i = \hat{v}^t_k. \tag{29}$$

In each time step, the state of each particle will be updated

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3045610, IEEE Access

according to a newly received RSSI data set, including the speed and moving direction of each particle. Thus, the trajectory of each particle will be updated accordingly. At the next step, we need to update weights of all particles.

3) Weight Update

Every time when a new set of RSSI data is received by the PF, the weight of each particle needs to be updated. The weight update for a particle i at time step t can be expressed by

$$w_i^t = \overline{w_i^{t-1}} \cdot g_i^t, \tag{30}$$

where g_i^t is a gain factor that can be derived by averaging the weight on each beacon receiver based on a Gaussian distribution [13], [19] as

$$g_i^t = e^{-\frac{1}{2}\left(\frac{\overline{D^t}}{\sigma_j}\right)^2}, \tag{31}$$

where σ_j is a measurement noise for a beacon receiver j and $\overline{D^t}$ is the mean distance difference between a particle i and a measurement of RSSI for each AP that can be calculated by

$$\overline{D^t} = \frac{1}{J}\sum_{j=1}^{J}(Z_{i,j}^t - \hat{Z}_{k,j}^t). \tag{32}$$

In (32), to better represent the distance differences, $\hat{Z}_{k,j}^t$ and $Z_{i,j}^t$ are the normalized distances, which can be expressed as

$$\hat{Z}_{k,j}^t = \hat{d}_{k,j}^t / \hat{d}_{max}, \tag{33}$$

$$Z_{i,j}^t = d_{i,j}^t / d_{max}, \tag{34}$$

where $\hat{d}_{k,j}^t$ is an estimated distance of tag k, $d_{i,j}^t$ is the exact distance of a particle toward AP j, $\hat{d}_{max}$ is the longest distance among estimated distances of a tag toward each AP, and d_{max} is the longest distance of all particles to one of the APs. Since a particle can be anywhere in a room, d_{max} can equal the diagonal distance of the room.

After obtaining the weights of all particles, we need to normalize them by

$$\overline{w_i^t} = \frac{w_i^t}{\sum_{i=1}^{N} w_i^t}, \tag{35}$$

such that the summation of all the weights $\sum_i \overline{w_i^t} = 1$. One reason of needing this normalization is that all the weights of particles can be scaled to a range between 0 to 1, which can help facilitate the convergence of the system. Another reason is for determining when to resample particles, which is discussed in the next step.

4) Resampling

A resampling can mitigate the degeneracy problem in a PF implementation. With weight $\overline{w_i^t}$ of each particle i at time t, we can resample all particles to reproduce a set of particles in which the particles with a higher weight have a higher chance to be reproduced. For doing so, we use the Stochastic Universal Sampling (SUS) approach [33] to resample particles. In our design, a resampling step is performed only when an estimated *effective sample size* $\hat{N}_{eff}$ is below a threshold N_T, and $\hat{N}_{eff}$ can be obtained by

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^{N}(\overline{w_i^t})^2}, \tag{36}$$

where N is the total number of particles and $\overline{w_i^t}$ is the normalized weight of a particle. The threshold N_T can be determined by $N \times \gamma$, where γ is a ratio coefficient.

When the $\hat{N}_{eff}$ is below the threshold N_T, a resampling algorithm is triggered. We use the SUS algorithm to resample particles, as shown in Algorithm 1. After executing this resampling algorithm, the weights of all particles shall be re-normalized to $1/|N|$.

At the final step, we will estimate the state of the tag by evaluating the states of all particles.

Algorithm 1 Stochastic Universal Sampling Algorithm

```
function SUS(X^t, W^t)        ▷ X^t is a set of particles generated at a previous step; W^t is the corresponding weight set of X^t.
    X̄^t ← ∅                   ▷ X̄^t is an empty set for storing new sampled particles.
    r ← rand(0, 1/|X^t|)      ▷ r is a random float number between 0 and 1/|X^t|.
    c ← w_1^t                 ▷ c is an initial weight of a particle.
    i ← 1                     ▷ i is an index of a particle.
    for k ← 1 to |X^t| do
        U ← r + (k − 1) · 1/|X^t|   ▷ U increases 1/|X^t| each step.
        while U > c do
            i ← i + 1         ▷ move to the next particle.
            c ← c + w_i^t     ▷ increase the weight summation by a new particle weight.
        end while
        X̄^t ← X̄^t ∪ {x_i^t}   ▷ x_i^t is included into the new particle set.
    end for
    return X̄^t
end function
```

5) Target State Estimate

Based on the estimated state $\hat{s}_i^t$ of each particle, the location of a tag k at time t can be approximated by averaging all the existing particles as:

$$\hat{P}_k^t = \frac{1}{N}\sum_{i=1}^{N} P_i^t. \tag{37}$$

In the next section, we evaluate our IPS by both simulations and experiments.

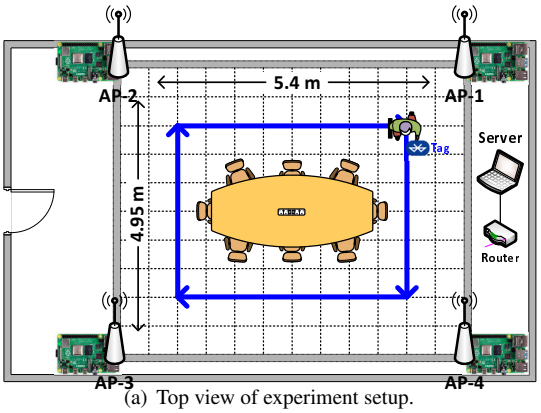
VI. PERFORMANCE EVALUATION

To verify the proposed IPS, we first evaluated the performance of the proposed system by simulations. Following that, we conducted several experiments by a testbed to see the system performance in real implementations.

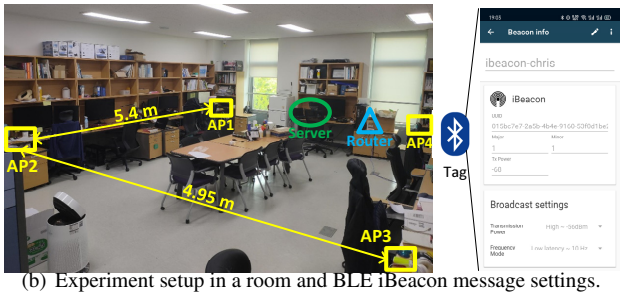
A. SIMULATION

We set up a simulation environment with an area 20 m×10 m by Python [34]. In this simulation, we place several APs at corners and boundary lines of a virtual room and one tag to be localized by default. For noises of RSSI data, we added a random variable following Gaussian distribution ($N(0, \sigma^2)$),

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3045610, IEEE Access



(a) Top view of experiment setup.



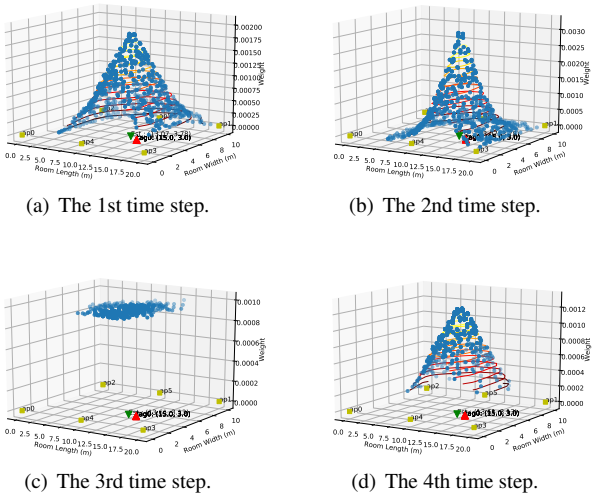
(b) Experiment setup in a room and BLE iBeacon message settings.

FIGURE 5: Experiment setup.

$\sigma = 1$ dBm by default) on the generated RSSI data. We obtained the results in each simulation that runs for more than 1.2 hours simulation time, and the target moving path is the same to the experiment shown in Section VI-B and Fig. 5(a). For the RSSI-to-distance mapping function, our simulations share the same parameters used in our experiments, which is described in Section VI-B1. The simulation parameter configuration is shown in Table 2. The baselines used to compare with our PFIPS are as follows:

- oIPS, the original IPS proposed in our previous paper and other papers [16], [17], [35].
- eSP, an extended version of the smart parking (SP) system [19] based on a particle filter.
- Tril, a conventional trilateration (or multilateration) approach [1], [24].

Fig. 6 shows a simulation process having 6 APs denoted by yellow rectangles on corners and each boundary lines and 1000 particles denoted by the scattered blue points. The red up-pointing triangle is the actual location of a tag, and the green down-pointing triangle is the approximated location of the tag by the current exiting particles. At the 1st time step of the simulation, every particle has an initial weight (i.e., probability) corresponding to the possibility of a location, and the variance of the particle distribution is large, as shown in Fig. 6(a). At the 2nd time step, the particles are refined by the weight update step described in Section V-C3 in which the weights of all the particles are updated based on a newly received RSSI data set, as shown in Fig. 6(b). At the 3rd time



(a) The 1st time step. (b) The 2nd time step.

(c) The 3rd time step. (d) The 4th time step.

FIGURE 6: Particle distribution with weights in a room having 6 APs at different time steps.

step, the effective particle count reaches a threshold, which triggers a resampling procedure described in Section V-C4 that filters out particles with a lower weight and normalizes others later, as shown in Fig. 6(c). At the next step shown in Fig. 6(d), the weights of all particles are updated again according to the current received RSSI data set. Through this kind of process, the particles are continuously updated and the position of the tag can be estimated.

TABLE 2: Simulation Configuration

Name	Content
Scene size	20 m×10 m
Particle set size	1000
AP set size	2, 4, 6, and 8; 6 by default
RSSI noise	$N(0, \sigma^2)$, $\sigma = 1, 3, 5$
σ in (31)	0.8
γ for (36)	0.8

1) Localization Error Comparison

First of all, we compare our PFIPS with the three baselines. Figs. 7 and 8 show the comparison of localization error among the schemes by simulations. Fig. 7 shows that localization error of our PFIPS outperforms that of our previous work oIPS and the conventional trilateration scheme, and slightly better than that of eSP on the mean and median values. Note that the orange line in each box represents a median of the localization error and the green up-pointing triangle is the mean value. The box in each case covers 50% of the data, including 25% of the data above and below the median, respectively. The maximum and minimum values to the median in each case are calculated by 1.5 times of the box coverage. So, the range from a bottom line to a top line accounts for more than 99% of all the data in a Gaussian model. The circles are outliers that are not in the previous range.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3045610, IEEE Access

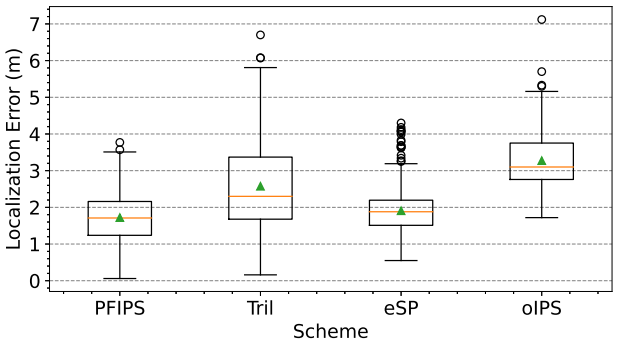


FIGURE 7: Localization error in comparison with other schemes.

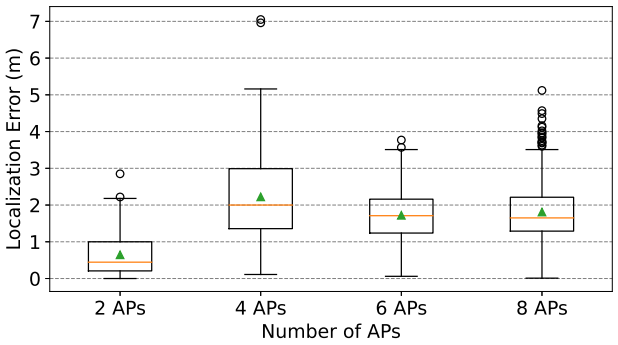


FIGURE 9: Localization error by different AP counts.

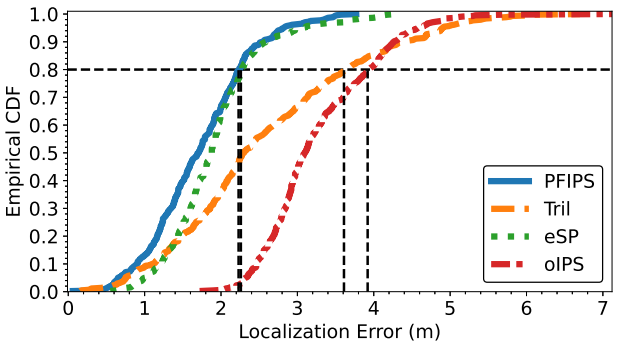


FIGURE 8: CDF of localization error in comparison with other schemes.

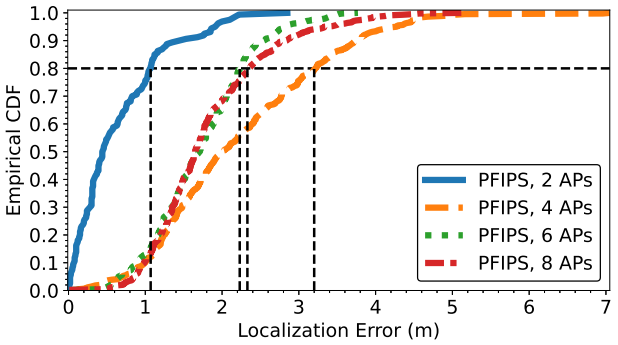


FIGURE 10: CDF of localization error by different AP counts.

Fig. 8 shows empirical Cumulative Distribution Function (CDF) of the schemes. Our PFIPS has a 2.21 m localization error for the 80th percentile based on our simulation data, whereas eSP reaches about 2.23 m localization error for the same percentile. The other two schemes, oIPS and Tril, reach about 3.96 m and 3.61 m, respectively. Although our PFIPS outperforms eSP at the lower localization error and shows a similar performance at the higher localization error, the experiment results shown later in Section VI-B4 demonstrate that our PFIPS can outperform the other schemes in a real world scenario.

2) Impact of Number of APs

Naturally, the more APs are available, the higher the localization accuracy can be. The number of APs limits the number of sources of RSSI data, thus it can affect the localization error. Therefore, we conduct simulations by varying the count of APs placed in the virtual room. We divide the APs into two groups and equally place them at each length of the virtual room, except the case of 2 APs in which the 2 APs are located at the two ends of the long symmetry line of the virtual room. When AP count is 2, the tag is positioned only in a 1D space, and the tag in other cases is positioned in a 2D space.

Figs. 9 and 10 show box plots and CDFs of the localization error, respectively. From Fig. 9, we can see that, except the case of 2 APs, as the count of AP increases, the localization error is reduced. For the case of 2 APs, since the positioning dimension is reduced, the localization error is also reduced, where the median of the localization error is about 0.5 m.

The CDFs of the localization errors shown in Fig. 10 present the overall influences of AP counts. For cases of 4, 6, and 8 APs, 80% of the localization error is within 3.21 m, 2.22 m, and 2.27 m, respectively. For the case of 2 APs, the error decreases to less than 1.05 m. Based on the obtained results, we can confirm that more available APs can improve the accuracy of tag positioning in our simulation. However, when the count of AP increases from 6 to 8, the localization error does not improve much. This implies that the localization error of our PFIPS may have a saturation level in which receiving more RSSI data from more APs can not reduce it anymore.

3) Impact of Noise (σ)

Since RSSI data are noisy, it is worthy to see the impact of different noise levels on the localization accuracy in our system. To analogize the noises, we add a Gaussian random number to each RSSI point, which follows $N(0, \sigma^2)$ distribution. Then, we vary σ from 1 to 5 with a step 2 to see the impact.

Fig. 11 shows box plots of the localization errors in different σ values. As the increasing of the σ, the localization error is also increased. Specifically, when σ increases from 1 to 5, the median of the localization error increases from about 1.7 m to 2.5 m. Meanwhile, we can also find that the mean values (shown by up-pointing triangles) deviate

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3045610, IEEE Access

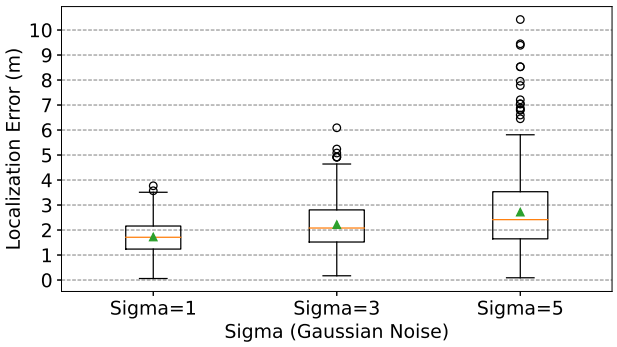


FIGURE 11: Localization error by different noise levels (σ).

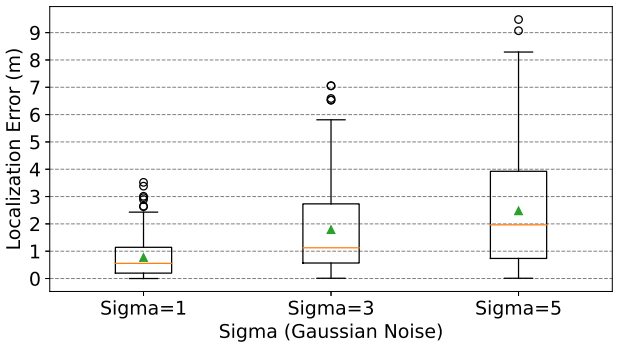


FIGURE 14: Localization error for 2 APs with different noise levels (σ).

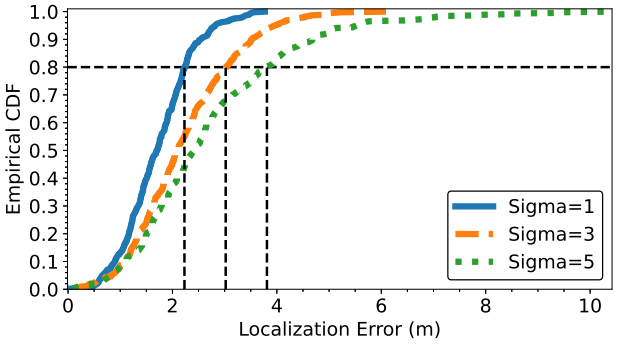


FIGURE 12: CDF of localization error by different noise levels (σ).

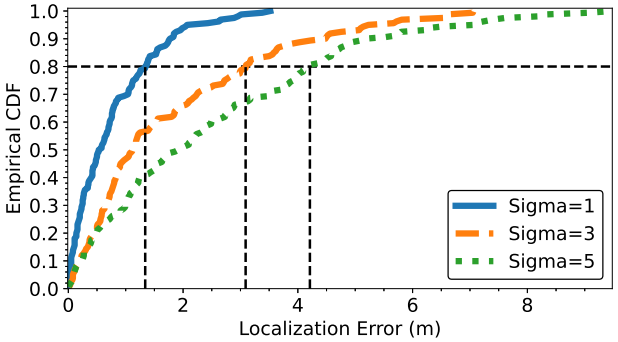


FIGURE 15: CDF of localization error for 2 APs with different noise levels (σ).

from the medians upwardly more and more, indicating that the increased noise level skews the data distribution. Fig. 12 shows the CDFs of localization errors with different σ values. When $\sigma = 1$, 80% of the localization error is below about 2.1 m. However, when σ increases to 5, the 80th percentile reaches to 3.8 m, and the tail extends to more than 10 m.

Fig. 13 exhibits convergences of the localization errors with confidence intervals on a scale of simulation time step. The dashed lines represent mean values of the localization errors. We can find that the converged mean localization errors for $\sigma = 1, 3, 5$ are about 1.1 m, 2.4 m, and 2.8 m at the end of the scale, respectively. We also notice that the confidence intervals increase along with the increase of σ, meaning that more uncertainties are expected.

We also particularly investigate the performance of the case of 2 APs with different σ values. Figs. 14 and 15 depict the impact of σ value in the 1D scenario. Similar to the previous 2D case, the increased noise levels downgrade the performance of the proposed PFIPS. When σ is 5, the mean and the 80th percentile of localization error are about 1.9 m and 4.25 m, respectively.

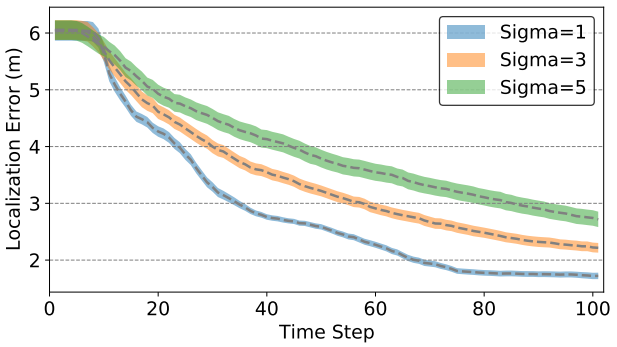


FIGURE 13: Localization error convergence.

B. EXPERIMENT

To evaluate performance of the proposed PFIPS in the real world, we set up a testbed having a similar setting to the simulations. Fig. 5 shows the experiment configurations. In this experiment, we use 4 Raspberry Pi4 boards as APs that connect to a server via an Ethernet. The server is configured to connect to a router, and the 4 APs can communicate with the server via the router for uploading beacon message information sent from a tag. Meanwhile, a tag in this experiment broadcasts BLE iBeacon messages [35] to the surrounding APs. The experiment configuration is shown in Table 3. Before running our PFIPS, we need to calibrate the RSSI-to-distance mapping function for our testbed; next, we verify the proposed KF-based smoothing module; eventually, we experiment our PFIPS to compare with the baselines in both 1D and 2D scenarios.

TABLE 3: Experiment Configuration

Name	Content
RSSI source	BLE iBeacons transmitted by a smartphone
Tx Bluetooth version	5.0
Rx Bluetooth version	5.0
Tx power	-56 dBm
Tx frequency	10 Hz
Server	A Samsung laptop with Ubuntu 18.04
AP	Raspberry Pi4
Tag	A smartphone running the Beacon Simulator v1.5.1 application
Localization area	5.4 m×4.95 m
System noise Q in KF	0.0005
Measurement noise R in KF	20
σ in (31)	0.5
γ for (36)	1.2

1) RSSI Mapping Calibration and Curve Fitting

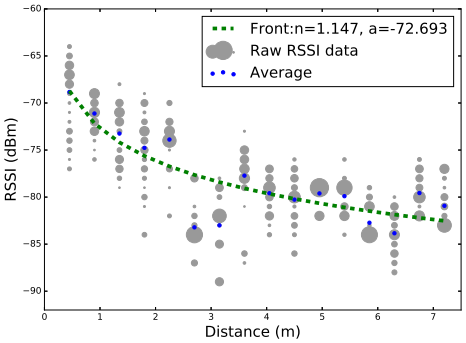
For calibrating the RSSI-to-distance mapping function for our testbed, a person holding a smartphone that broadcasts BLE beacon messages moves away from an AP in a fixed distance interval, and raw RSSI data are collected at every distance point. We totally collected 16 to 17 distance points in two cases, such as front-facing and back-facing. The front-facing case is measured when the person faces the AP while holding the smartphone in front of his/her chest, and the back-facing case vice versa. Fig. 16 shows two calibration cases of RSSI mapping.

At each measured distance point, we collected 300 raw RSSI data points for 10 minutes. A larger circle in Fig. 16 means more data points collected at the corresponding receiving power level. Based on averaged values of the data, we use a curve fitting function in the Python SciPy package [36] to find proper parameters in (21). The found parameters are $n = 1.147$ and $a = -72.693$ for front-facing case shown in Fig. 16(a), and $n = 0.329$ and $a = -80.596$ for back-facing case shown in Fig. 16(b). By observing the raw data of the back-facing case, the human body absorbed a quite amount of energy from the transmitted signals, causing a detrimental effect on the fitted curve. In a real deployment environment, the two cases may mix together showing more dynamic RSSI values. In the next subsection, we will show the verification of the RSSI data smoothing function in our IPS.

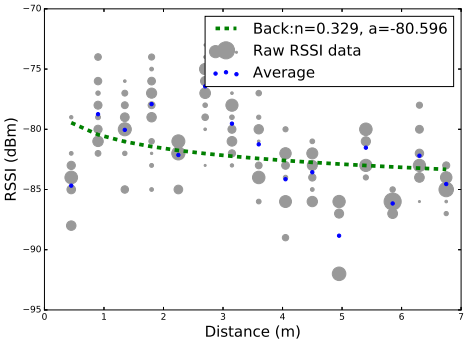
2) RSSI Data Smoothing

For verifying the effectiveness of the RSSI data smoothed by the proposed KF, we investigate the output data from the KF. Fig. 17 shows both RSSI data and its mapped distances on the basis of time step. The RSSI data in Fig. 17(a) have both raw and smoothed 500 data points using the measurement with blue color on the left-hand side. We can see that the raw RSSI data fluctuate between -80 and -64 dBm, and the smoothed ones are stabilized to the range between -72 to -68 dBm. The corresponding mapped distances are also shown in Fig. 17(a) based on the smoothed RSSI data, which have about 1 to 1.2 m variance compared with the ground truth distance (i.e., line with red color) that in this case is 1.8 m.

However, as shown in Fig. 17(b), when the BLE transmitter moves further away from the AP to 4.05 m, even though the raw RSSI data are denoised, the mapped distances still have a much larger variance from the ground truth distance.

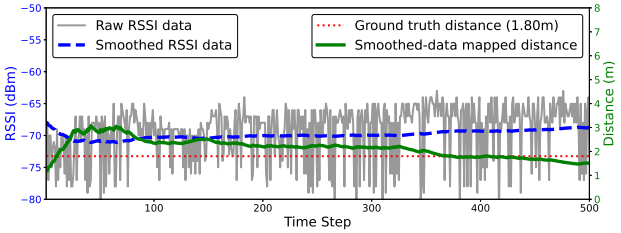


(a) Front-facing RSSI measurement and curve fitting.

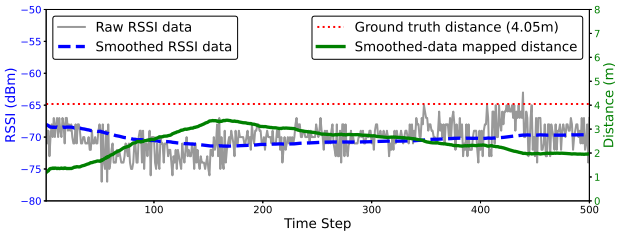


(b) Back-facing RSSI measurement and curve fitting.

FIGURE 16: RSSI measurement and curve fitting.



(a) Smoothed RSSI data and its mapping distance at 1.8 m.



(b) Smoothed RSSI data and its mapping distance at 4.05 m.

FIGURE 17: Raw RSSI data smoothed by the KF.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3045610, IEEE Access

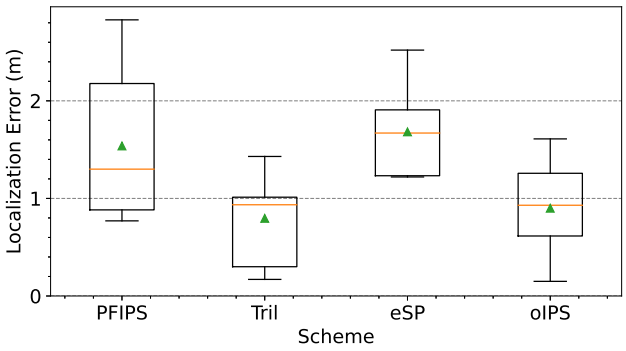


FIGURE 18: Localization error in 1D scenario experiment.

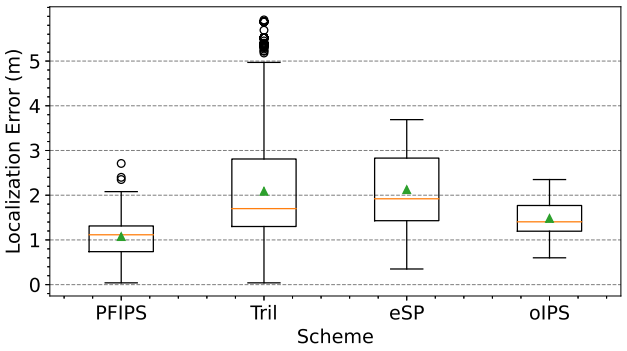


FIGURE 19: Localization error in 2D scenario experiment.

According to our distance mapping results, we notice that when the BLE transmitter is a bit far away from an AP, its mapped distances may suffer from a higher variance to the ground truth distance. This information indicates that, when a deployment density of APs is low, the positioning accuracy can decrease due to a higher chance that a BLE transmitter is relatively far away from APs.

3) 1D Scenario

Positioning a target in a 1D scenario, such as a corridor or a hallway, can happen quite often in the real world, so it is interesting to see the performance of our proposed PFIPS in this kind of scenarios. For experimenting this 1D scenario, we aligned 2 APs on a line with 6.3 m long. Fig. 18 shows the results of localization error in the 1D scenario. The median of the localization error of PFIPS in Fig. 18 is about 1.25 m, which is worse than that of Tril and oIPS approaches but better than that of eSP. The reason for the downgraded performance of our PFIPS can be that the less RSSI data from only 2 APs may provide insufficient information about the location belief of the tag. Another possible reason is that while we were conducting the experiments, there could be some intermittent interference sources that disturb our test. Surprisingly, the Tril approach gets a quite good performance, which suggests that for a 1D case, the trilateration approach can be a good candidate.

4) 2D Scenario

For a 2D scenario, we placed 4 APs in IoT Lab, Sungkyunkwan University, as shown in Fig. 5(b). We configured a smartphone to transmit BLE iBeacon messages via the Beacon Simulator application in which the parameters for transmitting the messages are set at a 10 Hz frequency (i.e., 1 message per 0.1 s) and at a transmission power -56 dBm, as shown in the right-hand side of Fig. 5(b). The localization area is 5.4 m long and 4.95 m wide. The σ and γ are set to 0.5 and 1.2, respectively, which is shown in Table 3. Note that there are many wireless access points and BLE devices running around and inside the experiment room, which can bring serious interference to the proposed system.

Figs. 19 and 20 show the performance of localization

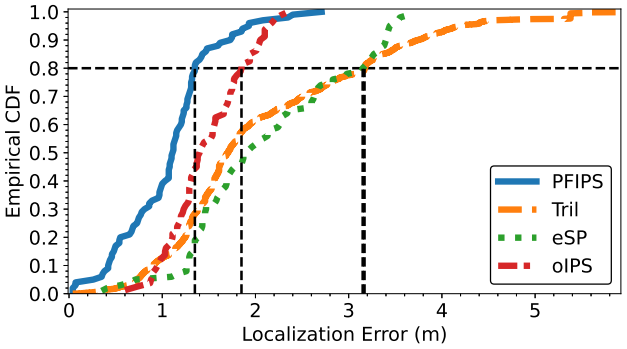


FIGURE 20: CDF of localization error in 2D scenario experiment.

errors in our experiments. The localization error of our PFIPS reaches less than 1.38 m for 80% of our experiment data, which improves 24.1% in comparison with oIPS. The experiment results that we obtained here demonstrate that our PFIPS outperforms the state-of-the-art schemes, though the simulation results reported in Section VI-A1 show a different performance in which our PFIPS is slightly better than eSP on localization error. The reason can be that in the simulations, we artificially add a Gaussian noise into RSSI data, but in reality the noises of the RSSI data sensed by the APs in our testbed may not follow a Gaussian distribution. These non-Gaussian noises can be caused by various factors, such as signal reflections/multipath, hardware specifications, and environment interference sources.

C. DISCUSSION

Our simulations and experiments used an area 20 m$\times$10 m and 5.4 m$\times$4.95 m, respectively. For a larger area, considering the effective communication range of BLE, our PFIPS can work well by partitioning the large area into many small cells and placing APs on the grid lines. For selecting a cell in which a tag is possibly located, one can employ a simple multilateration approach to coarsely estimate a location of the tag. With the estimated coarse location of the tag, we are able to determine a small cell to run our PFIPS. When a tag crosses several small cells, we can periodically run the coarse location estimation to check whether the tag

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3045610, IEEE Access

resides in the same cell or not for determining the need of reinitiating our PFIPS. The work in [22] has explored a set-valued box PF-based indoor localization approach. The work uses a learning-based coarse location estimation approach to roughly position a target. Then it operates a PF in a square area identified by the coarse location. The coarse location estimation approach in [22] can also be used to conduct a similar task for a large area in our PFIPS. Thus, for the localization of a tag in a huge space, such a set-valued box PF-based indoor localization approach can be naturally applied to our PFIPS scheme. The implementation of this combined localization scheme is left as future work.

To compare with the CSI-based approaches, we have summarized the differences shown in Table 1. Generally, a CSI-based indoor localization method uses the much richer environment information contained in the subcarriers of CSI data to sense targets. The subcarrier information has both amplitude and phase information of the radio signal. By the radio signal transmitted via multiple antennas, a receiver can decode changing features of a target. In this way, an indoor target can be localized. New research [20] has revealed that by using multipath information of wireless signals, a single WiFi transceiver with a MIMO function is able to track targets in an indoor environment. However, a CSI-based method may also suffer from the noises mingled with the subcarriers, and to denoise the subcarriers, an advanced signal processing technique is required, which may further increase the system complexity. Different from the CSI-based approaches, our PFIPS can be deployed by COTS devices with a small effort, as the testbed used in this paper. But our PFIPS requires multiple APs to be installed in a tracking area, which may increase the deployment costs.

VII. USE CASES

We have demonstrated that our PFIPS shows a better performance in comparison with other schemes in terms of localization accuracy. To apply the PFIPS to a real-world scenario, this section discusses use cases that can be built based on our PFIPS.

A. VEHICLE PARKING AND POWER CHARGING STATIONS NAVIGATION

For an indoor vehicle parking scenario, an entering vehicle needs to search for an available parking lot. Enough APs can be deployed by a parking service provider (PSP) in this parking area to serve for localization purposes. The vehicle equipped with an On-Board Unit (OBU) having one or more wireless communication interfaces can broadcast beacon messages. The deployed APs collect these beacons along with RSS data, and send them to the PSP. The PSP uses the information received to localize and guide the vehicle to move toward a parking lot. An electric vehicle (EV) may also need to search a power charging station in this parking area. Depending on options, the PSP can lead this EV to different types of charging stations, such as normal-speed charging and super-fast charging.

B. PERSONAL ACTIVITY MONITORING WITH WEARABLE DEVICES

The elderly people may wear various wearable devices for healthcare purposes, for instance, wearable fitness trackers, smart health watches, and wearable ECG (i.e., electrocardiogram) monitors. Usually these wearable healthcare devices (WHD) have a communication ability, which may be used for localization. For example, a doctor at a remote healthcare center may need to monitor a senior's activities. A WHD of the senior sends data packets having body biology information back to the healthcare center. With those data packets, the center can also localize the senior. If the senior stays at one position in his/her house for a long time, the doctor or a monitoring system by considering multiple information sources (e.g., body temperature, blood pressure, and electrocardiography) can determine to turn on an alarm for sending an emergency team to rescue the senior at an estimated position.

C. PUBLIC TRANSPORTATION WITH AUTOMATIC FAIR CHARGING

Currently urban public transportation systems in different nations charge passengers usually by a rechargeable smart card (e.g., T-money in South Korea, Clipper in the US Bay Area, EZ-Link in Singapore, and Suica in Japan) or an one-time ticket. For using urban subways or fast transits, a passenger needs to tag on his/her smart card and walk through a waist-high turnstile, and tag off when existing. A passenger may also tag on a smartphone with a near-field communication (NFC) function to pass through a turnstile. Although the current means to access the public transportation systems gives a great convenience to passengers, the turnstile admission process is still not efficient enough to deal with rush-hour traffic flows, which can cause a long line-up. A possible improvement can be that a system automatically charges passengers when they enter a transportation system. To do so, an IPS is a fundamental system for the purpose. For example, a passenger having a smartphone or a smart tag that broadcasts beacons can be automatically charged when (s)he moves into a public transportation waiting area.

VIII. LIMITATIONS

We have described our proposed PFIPS and showed the performance evaluation. In general, the proposed PFIPS has a few limitations as follows:

- It needs to place several APs on the corners and boundary lines of a room, and the locations of the APs shall be known in advance.
- The accuracy of position information of an AP can affect the positioning accuracy of a tag.
- The proposed PFIPS may show a worse performance when the background noise is severe, since our system still relies on the RSSI information collected from beacon messages.
- For using the proposed PFIPS in different indoor environments, the RSSI-to-distance mapping function may

need to be re-calibrated, since a new environment can change the mapping relationship between RSSI and distance.

In our experiment, we have found that occasionally the estimated location of a tag drifts from a more accurate location previously obtained. This may be caused by the accumulated noise error in both locations and orientations.

IX. CONCLUSIONS

This paper proposed a particle filter-based indoor positioning system to localize tags that can broadcast BLE beacon messages. To reduce fluctuations of RSSI data, we design a Kalman Filter to smooth those data. Based on the smoothed RSSI data, we propose a particle filter to conduct positioning tasks. The proposed particle filter uses a Gaussian kernel to compute probabilities of particles and an RSSI gradient model for the motion estimation. We implemented the proposed PFIPS in a testbed using COTS devices, which shows the applicability of our system. Through the simulations and experiments, the results from our PFIPS demonstrate a better performance. The localization error of our PFIPS in our experiment improves 24.1% compared with a state-of-the-art scheme. As future work, we will integrate other sensing means to our system to improve the localization accuracy, and especially we will investigate an advanced CSI-based deep learning approach for positioning a target.

X. ACKNOWLEDGEMENT

The authors thank Bien Aime Mugabarigira and Xiaohong Yu for helping the collection of the experiment data.

REFERENCES

[1] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," IEEE Commun. Surveys Tuts., vol. 21, no. 3, pp. 2568–2599, thirdquarter 2019.

[2] F. Gu, X. Hu, M. Ramezani, D. Acharya, K. Khoshelham, S. Valaee, and J. Shang, "Indoor Localization Improved by Spatial Context—A Survey," ACM Comput. Surv., vol. 52, no. 3, Jul. 2019.

[3] H. Lim, C. Park, and H. Myung, "Ronet: Real-time range-only indoor localization via stacked bidirectional lstm with residual attention," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, pp. 3241–3247.

[4] C. Tao, Z. Gao, J. Yan, C. Li, and G. Cui, "Indoor 3d semantic robot vslam based on mask regional convolutional neural network," IEEE Access, vol. 8, pp. 52 906–52 916, 2020.

[5] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-based User Location and Tracking System," in Proceedings of IEEE INFOCOM 2000. The 19th Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, 2000, pp. 775–784.

[6] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool Release: Gathering 802.11n Traces with Channel State Information," SIGCOMM Comput. Commun. Rev., vol. 41, no. 1, p. 53, Jan. 2011.

[7] Y. Xie, Z. Li, and M. Li, "Precise Power Delay Profiling with Commodity WiFi," in Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, ser. MobiCom '15. New York, NY, USA: ACM, 2015, p. 53–64.

[8] L. Chen, I. Ahriz, and D. Le Ruyet, "AoA-Aware Probabilistic Indoor Location Fingerprinting Using Channel State Information," IEEE Internet of Things Journal, vol. 7, no. 11, pp. 10 868–10 883, 2020.

[9] D. Dardari, P. Closas, and P. M. Djurić, "Indoor Tracking: Theory, Methods, and Technologies," IEEE Trans. Veh. Technol., vol. 64, no. 4, pp. 1263–1278, 2015.

[10] M. Scherhäufl, M. Pichler, E. Schimbäck, D. J. Müller, A. Ziroff, and A. Stelzer, "Indoor Localization of Passive UHF RFID Tags Based on Phase-of-Arrival Evaluation," IEEE Trans. Microw. Theory Tech., vol. 61, no. 12, pp. 4724–4729, 2013.

[11] R. Faragher and R. Harle, "Location Fingerprinting With Bluetooth Low Energy Beacons," IEEE J. Sel. Areas Commun., vol. 33, no. 11, pp. 2418–2428, 2015.

[12] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," IEEE Trans. Signal Process., vol. 50, no. 2, pp. 174–188, 2002.

[13] J. L. Carrera Villacrés, Z. Zhao, T. Braun, and Z. Li, "A Particle Filter-Based Reinforcement Learning Approach for Reliable Wireless Indoor Positioning," IEEE J. Sel. Areas Commun., vol. 37, no. 11, pp. 2457–2473, 2019.

[14] Core Specification Working Group, Bluetooth SIG. Bluetooth Core Specification, version 5.2. (accessed: 09.14.2020). [Online]. Available: https://www.bluetooth.com/specifications/bluetooth-core-specification/

[15] R. Ayyalasomayajula, D. Vasisht, and D. Bharadia, "BLoc: CSI-Based Accurate Localization for BLE Tags," in Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies, ser. CoNEXT '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 126–138.

[16] Y. Shen, J. Kim, Y. Ahn, and J. P. Jeong, "Indoor Positioning System for Beacon Tag Tracking," in KICS Winter Conference 2020. The Korean Institutes of Communications and Information Sciences (KICS), Feb. 2020, pp. 480–481.

[17] B. Hwang and J. Jeong, "Design and Implementation of Indoor Positioning System using Particle Filter based on Wireless Signal Intensity," Journal of KIISE, Apr. 2020.

[18] M. Kim, D. Han, and J. K. K. Rhee, "Machine Learning for Practical Localization System Using Multiview CSI," IEEE Access, vol. 8, pp. 184 575–184 584, 2020.

[19] A. Mackey, P. Spachos, and K. N. Plataniotis, "Smart Parking System Based on Bluetooth Low Energy Beacons With Particle Filtering," IEEE Systems Journal, pp. 1–12, 2020.

[20] C. Wu, F. Zhang, B. Wang, and K. J. R. Liu, "EasiTrack: Decimeter-Level Indoor Tracking With Graph-Based Particle Filtering," IEEE Internet of Things Journal, vol. 7, no. 3, pp. 2397–2411, 2020.

[21] R. Ayyalasomayajula, A. Arun, C. Wu, S. Sharma, A. S. D. Vasisht, and D. Bharadia, "Deep learning based wireless localization for indoor navigation," in The 26th Annual International Conference on Mobile Computing and Networking, ser. MobiCom '20. New York, NY, USA: Association for Computing Machinery, 2020.

[22] P. Malekzadeh, A. Mohammadi, M. Barbulescu, and K. N. Plataniotis, "STUPEFY: Set-Valued Box Particle Filtering for Bluetooth Low Energy-Based Indoor Localization," IEEE Signal Process. Lett., vol. 26, no. 12, pp. 1773–1777, 2019.

[23] J. P. Jeong, S. Yeon, T. Kim, H. Lee, S. M. Kim, and S.-C. Kim, "SALA: Smartphone-Assisted Localization Algorithm for Positioning Indoor IoT Devices," Wirel. Netw., vol. 24, no. 1, p. 27–47, Jan. 2018.

[24] J. Röbesaat, P. Zhang, M. Abdelaal, and O. Theel, "An Improved BLE Indoor Localization with Kalman-Based Fusion: An Experimental Study," Sensors, vol. 17, no. 5, 2017.

[25] J. Jun, L. He, Y. Gu, W. Jiang, G. Kushwaha, V. A., L. Cheng, C. Liu, and T. Zhu, "Low-Overhead WiFi Fingerprinting," IEEE Trans. Mobile Comput., vol. 17, no. 3, pp. 590–603, Mar. 2018.

[26] S. Xu, R. Chen, Y. Yu, G. Guo, and L. Huang, "Locating Smartphones Indoors Using Built-In Sensors and Wi-Fi Ranging With an Enhanced Particle Filter," IEEE Access, vol. 7, pp. 95 140–95 153, 2019.

[27] G. Guo, R. Chen, F. Ye, X. Peng, Z. Liu, and Y. Pan, "Indoor Smartphone Localization: A Hybrid WiFi RTT-RSS RangingApproach," IEEE Access, vol. 7, pp. 176 767–176 781, 2019.

[28] J. P. Jeong, Y. Shen, S. Kim, D. Choe, K. Lee, and Y. Kim, "DFC: Device-Free Human Counting through Wi-Fi Fine-Grained Subcarrier Information," IET Communications, vol. 14, pp. 1–13, Oct. 2020.

[29] D. Simon, Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches. USA: Wiley-Interscience, 2006.

[30] C. Andrieu, N. D. Freitas, A. Doucet, and M. I. Jordan, "An Introduction to MCMC for Machine Learning," Machine Learning, vol. 50, no. 1-2, pp. 5–43, 2003.

[31] R. Bardenet, A. Doucet, and C. Holmes, "On Markov Chain Monte Carlo Methods for Tall Data," Journal of Machine Learning Research, vol. 18, no. 47, pp. 1–43, 2017.

[32] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, and G. Luo, "Smartphone-Based Real Time Vehicle Tracking in Indoor Parking Structures," IEEE Trans. Mobile Comput., vol. 16, no. 7, pp. 2023–2036, 2017.
[33] J. E. Baker, "Reducing Bias and Inefficiency in the Selection Algorithm," in Proceedings of the Second International Conference on Genetic Algorithms and Their Application. USA: L. Erlbaum Associates Inc., 1987, p. 14–21.
[34] Python. (accessed: 09.15.2020). [Online]. Available: https://www.python.org/
[35] F. Zafari, I. Papapanagiotou, M. Devetsikiotis, and T. Hacker, "An iBeacon based Proximity and Indoor Localization System," 2017.
[36] SciPy. (accessed: 09.15.2020). [Online]. Available: https://www.scipy.org/

YIWEN SHEN (S'15) received the B.S. degree from the Department of Communication Engineering, North University of China (NUC), China in 2009, and the M.S. degree from the Department of Mechatronics Engineering, NUC, China in 2013. He is currently pursuing the Ph.D. degree with the Department of Electrical & Computer Engineering, Sungkyunkwan University, South Korea. His research interests include intelligent transportation systems, vehicular networks, IoT systems, indoor localization, and wireless communications. He received the China State Scholarship from the China Scholarship Council. He is a student member of IEEE and IEEE Communication Society.

BEOM HWANG (S'20) received the B.S. degree from the School of Biomedical Engineering at Jeonbuk National University in 2014, and the M.S. degree from the Department of Computer Science and Engineering at Sungkyunkwan University in 2019. His research interests include Internet of things and indoor localization. He is a student member of IEEE.

JAEHOON (PAUL) JEONG (M'04) is an associate professor in the Department of Computer Science and Engineering at Sungkyunkwan University in Korea. He received his Ph.D. degree in the Department of Computer Science and Engineering at the University of Minnesota in 2009. He received his B.S. degree in the Department of Information Engineering at Sungkyunkwan University and his M.S. degree from the School of Computer Science and Engineering at Seoul National University in Korea in 1999 and 2001, respectively. His research areas include cyber-physical systems, Internet of Things, indoor localization, vehicular networks, software-defined networking, network functions virtualization, and network security. His two data forwarding schemes (called TBD and TSF) for vehicular networks were selected as spotlight papers in IEEE Transactions on Parallel and Distributed Systems in 2011 and in IEEE Transactions on Mobile Computing in 2012, respectively. Dr. Jeong is a member of ACM, IEEE and the IEEE Computer Society.