IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 13, 2021, accepted February 24, 2021, date of publication March 1, 2021, date of current version March 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3062818

Adaptive Sequential Monte Carlo Filter for Indoor Positioning and Tracking With Bluetooth Low Energy Beacons

F. SERHAN DANİŞ[1,2], **A. TAYLAN CEMGİL**[1], **AND CEM ERSOY**[1], **(Senior Member, IEEE)**

[1]Department of Computer Engineering, Boğaziçi University, 34342 Istanbul, Turkey
[2]Department of Computer Engineering, Galatasaray University, 34349 Istanbul, Turkey

Corresponding author: F. Serhan Daniş (serhan.danis@boun.edu.tr)

This work was supported by the Turkish Directorate of Strategy and Budget through the TAM Project under Grant 2007K12-873.

ABSTRACT We model the tracking of Bluetooth low-energy (BLE) transmitters as a three layer hidden Markov model with joint state and parameter estimation. We are after a filtering distribution by Bayesian approximation using Monte Carlo sampling techniques. In a test environment decorated with multiple BLE sensors, the tracking relies only on the naturally unreliable received signal strength indicator (RSSI) of the captured signals. We assume that the tracked BLE transmitter does not provide any other motion or position related information. Hence, the transition density is designed to be merely a diffusion where the probability measures are diffused into the neighboring space. This makes the diagonal error covariance factor of the prediction density, namely the diffusion factor, the most important parameter to be tuned on the fly. We first show an experimental proof of concept using synthetic data on real trajectories by comparing three parameter estimation approaches: static, decaying and adaptive diffusion factors. We then obtain the results on real data which show that online parameter sampling adapts to the observed data and yields lower error means and medians, but more importantly steady error distributions with respect to a large range of parameters.

INDEX TERMS Bluetooth low-energy, indoor positioning and tracking, parameter estimation, sequential Monte Carlo, wasserstein interpolation.

I. INTRODUCTION

This work addresses the problem of positioning and tracking in indoor environments using the probabilistic sampling methods with Bluetooth Low-Energy (BLE) signal indicators as measurements. In a closed indoor area decorated with multiple BLE sensors, we track the position of a mobile beacon that transmits only BLE messages but no other information about its movement or its whereabouts. By the nature of radio-frequency signals, message transfer is affected by the factors like occlusion, scattering, reflections and interference from other devices, resulting with the multipath phenomenon, mostly because of furnitures and materials used in the architecture of the building.

A. INDOOR POSITIONING

Tracking of objects indoors in real-time has become essential in many fields such as retail, logistics, marketing and health. With the aging population of the world, health workers need assistive systems for tracking elderly or sick people in order to mitigate their workload [1]. The Covid-19 pandemic has introduced the contact tracing notion, which can also be resolved with accurate positioning systems [2]. In marketing and retail, these systems help make targeted and location based promotions or advertisements by tracking the client behaviors [3]. In logistics and industrial environments, indoor positioning has become a need in asset and shipment tracking [4] and even for livestock tracking [5].

Indoor positioning systems (IPS) are formed of a group of appareils used to estimate the micro-locations of the objects or the people. Target locations are closed environments where satellite based positioning systems yield highly inaccurate estimations or fail entirely due to the coverage

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Seo Kim.

loss of their signals. With the lack of satellite positioning systems, there appeared diverse modalities, various signal parameters/statistics and associated processing technologies adapted to IPS.

Visual information provides the most accurate solutions for positioning, but requires high amounts of data to be captured, transferred and processed. The domain has developed several processing algorithms like visual odometry and scanmatching that use conventional cameras [6] and depth cameras [7]. Visible light communication that use LED beacons is specially designed for indoor positioning [8], but lately, with the introduction of multiple augmented reality (AR) detectors, AR tags have also become robust alternatives for visual information processing [9].

Ultrasound based positioning systems provide accurate localization at low cost. Such systems use the time-of-flight (TOF) of sound to estimate the distances between a mobile receiver and stationary transmitters. Estimations are then multilaterated to obtain a position [10]. The same phenomenon that uses light modality is commercialized into range finders and lidars. These systems are specialized in finding the occluded areas or objects in their ranges, making them high precision, but expensive indoor positioning alternatives [11].

Whereas vision, inertial navigation and radio-frequency (RF) based data processing solutions coexist in hybrid systems with the pervasiveness of smart phones [12], low power consumption and low processing requirements make the pure RF based systems preferable. Wi-Fi technology is one of the oldest RF technologies used for indoor positioning because of its easy-to-use and already installed infrastructure for accessing local networks. Signal parameters and preprocessing techniques prior to position estimations depend mostly on the experiments performed with Wi-Fi infrastructures in the first decade of this millenium [13]. A radio-frequency identification (RFID) based positioning system consists of two types of appareils: readers and transponders or tags, which may in turn be passive or active. Cheap passive tags are powered by the radio waves emitted by the reader [14], hence narrowing the emission distance under one meter, whereas active tags have their own energy sources and can be read from greater ranges up to hundreds of meters [15]. Ultra wideband is one of the emerging technologies used for indoor positioning. With its high data rate which can attain the speeds of up to 100 Mbps, it has become a good solution for near-field data transmission [16].

BLE transmitters are becoming more ubiquitous everyday, because of their low energy consumption, high accessibility and small mass. A properly configured transmitter can stand online from months to years with conventional batteries. These transmitters can communicate easily with contemporary handheld devices, making them readable by any BLE-ready device [17]. Transmitters are also called beacons, as they were first thought as stationary objects that broadcast Bluetooth packets. Recently, thanks to their small volumes and masses, they are becoming the actual mobile objects of interest that are being tracked by the help of the surrounding sensor systems. These cheap devices are attached to items, assets, animals or people that need to be tracked.

For the RF based positioning systems, there exist a handful of approaches of signal parameters, the most used of which are received signal strength indicator (RSSI), time of arrival (TOA), time difference of arrival (TDOA), angle of arrival (AOA) and channel state information (CSI). The downside of RF signals is that they do not work robustly because of the multipath problems. Therefore, captured data need to be preprocessed before the actual filtering or estimations of positions are performed [13]. Two of these preprocessing techniques are trilateration and fingerprinting. Trilateration is the estimation of positions given the distances between the tracked object and static landmarks whose positions are priorly known. The technique directly depends on the accuracy of these distances, which fluctuates as the signal parameters fluctuate [18]. In one specific work that uses the trilateration technique, Cantón Paterna *et al.* track a mobile sensortag using the surrounding sensing computers [19]. They employ multiple BLE channels to mitigate the effect of fast fading and interferences. The positions are estimated through a trained trilateration method based on weights and smoothed via Kalman filtering. They reach an error rate of 1.82 m in a 54 m^2 room, and 4.6 m in a 290 m^2 room.

Fingerprinting, also called prior scene analysis or RF signal profiling, generates a RF map of the region of interest. Basically, an IPS system estimates the positions by comparing the live measurements with the fingerprints. The accuracy depends on the profiling density of the corresponding parameter, causing a trade-off between the installation overhead versus the precision [20]. Fingerprinting techniques have been shown to be more accurate with respect to trilateration techniques [21]. Similarly, in our experimental setup, we want to estimate and track the position of a transmitter that emits BLE packets. These packets are captured and corresponding RSSI measurements are generated by the sensors stationed in the same indoor environment. The locations of the sensors are not necessarily known as we rely on a fingerprinting technique, but a good coverage is obligatory.

B. INFERENCE WITH MONTE CARLO SAMPLING

We model the system as a state space model (SSM), in which we observe RSSI measurements and estimate the positions. The transition densities on the latent chains are assumed to be Gaussian and gamma. The emission density, that is the distribution of the RSSI values conditioned on the positions, is estimated using a Wasserstein interpolation technique [20]. We attempt to approximate the nonlinear Bayesian filtering using the sequential Monte Carlo filter. Sequential Monte Carlo filters, or namely particle filters or bootstrap filters, are well known for their efficiency in solving numerically complex and non-standard problems that were previously intractable [22]. Moreover, two important properties of the proposed system restrict us from using linearizable and exact methods: the emission densities that map the position readings to the measured RSSI values are nonlinear and non-Gaussian, and their indices are also discrete integers.

Monte Carlo sampling techniques have been long used in positioning and have a popularity in both combining multiple sensor readings like visual information and inertial information and perform better against its counterparts like Kalman filter family, especially for nonlinear approximation. They are known to be efficient and easily implemented [23]. Sequential Monte Carlo (SMC) filters have exclusive use cases in indoor positioning with RF signals. Nurminen *et al.* [24] build a particle filter where pedestrian dead-reckoning (PDR) predicts the position particles, and WiFi RSSI measurements and the floor plan correct the predictions and achieve a median error rate of 0.8 meters. Shubair and Elayan [25] propose a particle filter based technique for Wireless Sensor Networks (WSN) that uses TDOA parameter as measurements. Their TDOA-PF technique is shown to outperform the other well known techniques with respect to both accuracy and robustness. Filipek and Kovarova [26] use BLE transmitters as signal sources and estimate the distances between several of these stationary transmitters and use this information for weighing the particles of the generic SMC filter. The tests are performed with previously and discretely selected positions, and the method achieves an error rate of 0.35 meters in average.

Adaptivity in SMC filters may serve for adjusting different parameters, in order to mitigate the estimation errors in changing conditions in the dynamics or in the environment. Lang *et al.* [27] design an adaptive particle filter that is augmented with two resampling steps, one before the prediction step and the other before the weight update step, so that the number of particles is decreased in the second resampling step which in order decreases the computation time. The tests are performed on a real robot that uses sonar readings as measurements. As another example, the size of the samples may be adapted on the fly by measuring the approximation error. In [28], the approach bases on updating the importance weights until the Kullback-Leibler (K-L) divergence between the maximum likelihood estimate and the underlying posterior is under a predetermined bound. The efficiency is boosted by avoiding unnecessary resamplings.

Despite their popularity, the SMC techniques have an important disadvantage: it is unclear how to set the noise parameters of the conditional distributions in the first place. Efficient adaptive mechanisms are needed to tune these parameters while performing the actual tracking operation. In [29], an adaptive noise variance selection strategy was designed. They obtain a blindness value that is controlled by two parameters: α for steepness and β for the position of the transition. The noise variances are adjusted according to the blindness estimation at the previous iteration, so that noise in the measurements are never amplified due to the dynamic components. Whereas our approach is similar to this work, we cannot control the emission density noise variances.

Auxiliary particle filters (APF) are proposed to solve the problem of adaptation and imprecisement of the sequential importance resampling (SIR) method due to a large variance of the particle weights [30]. Our proposed algorithm may also be associated with APF because of two stage particle assignments, but differs by the type of particle association between two stages. In APFs, firstly auxiliary variable indices are sampled with respect to some auxiliary likelihood values and the second stage particles are sampled using these indices. In our proposed method, we do not analytically link the first stage particles to the second stage, but sample second stage particles using the first stage particles as parameters. When weights are assigned for the second stage particles, they are also propagated to the first stage. This approach filters a trajectory of positions and noise parameters jointly.

When parameter estimation is the purpose of the SMC methods, we generally consider maximum likelihood estimation (MLE) of static parameters [31]. The main objective is to estimate an unknown static parameter from the data in an online or offline manner [32], however, we want the noise parameter to be not only statically estimated, but also to be adapted with respect to the changing conditions or properties, like the velocity of the tracked object or noise fluctuations in the measurement data. A common approach for the SMC filters to adapt to the changing and/or unknown motion of the tracked targets is to augment the state with a parameter under the name augmented particle filters, where the parameter to be estimated is moved to a new process and parameter estimations can be performed through this new process if sufficient data can be supplied [33].

One of the similar methods is the iterated batch importance sampling (IBIS) method that updates a grid of samples in the parameter space [34], but IBIS is not recursive and requires to compute probability density function of every measurement variable given the previous measurement in closed form. Papavasiliou [35] addresses the same problem using two layers of Monte Carlo methods to solve the previous recursivity issue. The first of these layers generates random grids of parameters, and a particle filter is run with each of these first layer parameters. Chopin *et al.* [36] extend the IBIS method to a grid of samples in the parameter space that are updated over time. At each step, a particle filter is used to process the measurement, and a new grid is generated. These methods involve nested layers of particle filters [37]. The nested methods have not yet been used for tracking of objects, nor for indoor positioning domain.

C. PROPOSED WORK

In SMC filters, the prediction step generally tries to model the dynamics of the tracked object by inserting some motion related data. In our model, we contrarily assume that we have very little information on how the object moves: it may only jump onto the neighboring coordinates. This "close" jumping behavior is defined by a normal distribution, which may also be named as a *diffusion*: the particles are diffused into the environment. A diagonal noise covariance is driven by a diffusion factor, which can also be indirectly associated with the velocity of the tracked object, thus becoming very essential for the SMC filter for accurate position estimation. When this diffusion factor is set to a high value, the system tends to be in an exploratory behavior, that is, the predictions are diffused rapidly over a wide area. This behavior increases the possibility of capturing the highest likelihood, but conversely the particle set will be distracted by spuriously similar patches that are highly probable in the RF map, and they will be updated accordingly, probably increasing the error. If an excessively low diffusion factor is chosen, the predictions tend to lag or get totally stalled and the estimated trajectory cannot catch the true trajectory. These caveats may lead to a divergence in accurate positioning and temporary but relatively highly erroneous position estimations.

In a diffusion based transition density, the diffusion factor should be optimized for each trajectory, but an optimum static parameter can only be properly estimated after at least one full run of the algorithm. Moreover, a *static diffusion* factor optimization approach may be misleading as the velocity of the object or the emission noise can vary throughout an experiment. We also show that a *decaying diffusion* factor may not be as accurate as a fine tuned constant one. We solve this issue by sampling the diffusion factor along with the positions: *adaptive diffusion* factor.

The contribution of this work is the on-the-fly estimation of the variable diffusion factor through importance sampling alongside the position estimations. This is achieved with a novel version of the nested SMC filter specially tailored for indoor positioning and tracking. We have two layers of particles, one for the positions and the other for the diffusion parameters. The weights evaluated using the measurements at each step are used firstly to estimate a weighted mean of positions. Secondly, the weights are propagated through the diffusion layer and a weighted diffusion factor can also be computed. We show that an adaptive diffusion factor is robust to inaccurate temporary estimations. It increases necessarily as the position estimations are degraded, and decreases with better position estimations. The results also show that an adaptive diffusion factor leads to better position estimations for BLE-based object tracking in indoor environments.

In the organization of the paper, Section II presents the details of the proposed approaches. Section III describes the test environment and experimental design. The experiments and results are discussed in Section IV, and we finalize our paper with conclusions and future options in Section V.

II. METHODOLOGY

A. TRACKING MODEL

We model the problem of "indoor positioning and tracking with BLE beacons" as a hidden Markov model (HMM), in which the diffusion parameters constitute a prior latent layer and the positions form a secondary one, and the RSSI measurements captured by the distributed sensors are the observations. We write the generative dependency model as follows:

$$\begin{aligned} k_0 &\sim p(k_0) \\ k_t &\sim p(k_t|k_{t-1}) \\ \mathbf{x}_0 &\sim p(\mathbf{x}_0) \\ \mathbf{x}_t &\sim p(\mathbf{x}_t|\mathbf{x}_{t-1}, k_t) \\ y_t^s &\sim p(y_t|\mathbf{x}_t) \end{aligned} \tag{1}$$

The diffusion layer forms a Markov chain, with a prior distribution at the initial time, $p(k_0)$. The diffusion parameter, k_t, at discrete time, t, is a function of the diffusion parameter at its previous time, $p(k_t|k_{t-1})$. The secondary latent layer is initialized likewise with a prior distribution, $p(\mathbf{x}_0)$, and the transition density at the current time is conditioned on the position at its previous time and the diffusion parameter at the current time, $p(\mathbf{x}_t|\mathbf{x}_{t-1}, k_t)$. We observe only the RSSI measurements, y_t^s, captured by a sensor, s. RSSI data are modeled by the emission density $p(y_t^s|\mathbf{x}_t)$. The problem forms the hidden Markov model shown in FIGURE 1. We want to make a Bayesian approximation to a filtering distribution $p(\mathbf{x}_t, k_t|y_{1:t}^s)$, and jointly find accurate position estimates of the tracked object and the diffusion factor given the measurements, or the RSSI values, so far.

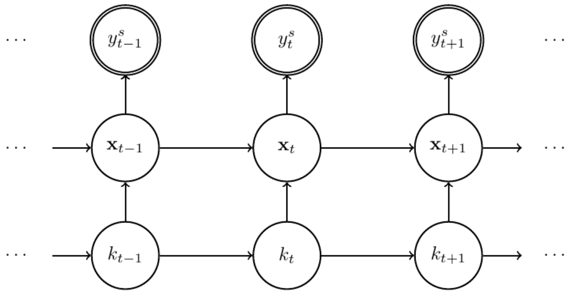


FIGURE 1. Proposed HMM with a latent chain for positions that depends on another one for diffusion factors.

1) TRANSITION DENSITY MODEL

The tracked object is a beacon that emits Bluetooth low energy packets. We restrict our beacon to reside on a plane, in which the position is composed of two cartesian coordinates, $\mathbf{x} = (x, y)$. We assume that these beacons do not have any other sensor or processing appliances, hence, we (assume to) have no access to any internal information about the movement of the tracked object, but we also assume that the object moves smoothly in the area so that its current position is directly related to its position at the previous timestamp (no kidnapping). This transition is modeled as a *diffusion* density which is described by a normal distribution with the previous position $\mathbf{x}_{t-1}$ as its mean, and a 2 × 2 covariance matrix that is driven by the diffusion factor, $k_t\mathbf{I}_2$.

$$p(\mathbf{x}_t|\mathbf{x}_{t-1}, kt) = \mathcal{N}(\mathbf{x}_{t-1}, k_t\mathbf{I}_2) \tag{2}$$

The diffusion factor serves as a scale for the noise covariance of the position chain. Therefore, this factor should be positive. To guarantee to easily sample positive diffusion factors, we define the transition density by a gamma distribution. The shape and scale parameters of the gamma distribution, a and b, are tuned accordingly to ensure that the diffusion factor at the previous timestamp, k_{t-1} is put at its mode, and a *sensitivity* value, ν, at its variance:

$$\begin{aligned} p(k_t|k_{t-1}) &= \mathcal{G}(a_t, b_t) \\ \text{with } a_t &= \frac{(k_{t-1})^2}{\nu} + 1 \\ \text{and } b_t &= \frac{\nu}{k_{t-1}} \end{aligned} \tag{3}$$

2) EMISSION DENSITY MODEL

The system depends solely on the RSSI measurements, y_t^s, captured by a sensor, s, at a discrete time step, t. To obtain a probability density of the RSSI measurements given the positions, namely a probabilistic radio map, we use the histograms of RSSI measurements, or fingerprints, previously captured on several positions in the test environment. Because we require to evaluate any RSSI measurement on any point, we utilize a histogram interpolation method, Affine Wasserstein Combination (AWC) [20]. From the two specific types of these interpolation methods, we choose Two-point Affine Wasserstein Combination which considers the closest and the most aligned two fingerprint positions to compute a histogram on the position $\mathbf{x}_t$.

$$p(y_t^s|\mathbf{x}_t, \Theta) = \tilde{h}(y_t^s; \mathbf{x}_t, \alpha, \beta) \tag{4}$$

where $\tilde{h}()$ denotes an estimate for the histogram of RSSI measurements for a specific sensor, s, at position $\mathbf{x}_t$. The parameters α and β are algorithm specific parameters. The algorithm is summarized in Section II-B3 as a part of SMC filters.

B. SMC FILTER FOR BLE LOCALIZATION

We first describe a generic SMC filter for fusing BLE data captured by several sensors. In this generic version, the diffusion factor is set as static and decaying, so the diffusion layer is replaced with a single static value or a deterministic decaying function for the current section. We evolve the setup into the adaptive SMC filter by using the diffusion transition density in Section II-C.

In the SMC formalization, N will denote the particle population size at each step of the SMC algorithm. The population size will always be constant in an experiment, but may be varied between experiments. We will denote each particle with $(\mathbf{x}_t^{(i)}, \omega_t^{(i)})$ where $\mathbf{x}_t^{(i)}$ is a 2D cartesian vector that stands for the position of the particle $i \in [1..N]$ at discrete time step $t \in [0..T]$ and $\omega_t^{(i)}$ is its current weight. Time indices will be put in subscripts and particle indices will be in parentheses in superscripts. Vectors will be shown with bold and lowercase characters, matrices are bold and uppercase. The predictions are marked with a tilde, $\tilde{}$.

1) INITIALIZATION

Initially, we have no information on the whereabouts of the transmitter. The particles of the initial population are modeled as continuously uniformly distributed.

$$\mathbf{x}_0^{(i)} \sim \mathcal{U}(\mathcal{A}) \tag{5}$$

where $\mathcal{A}$ is the rectangular area of the test environment that supports the uniform distribution.

2) IMPORTANCE SAMPLING

For an appropriate proposal distribution, we set a diffusion based density to sample new particles from, as stated in Section II-A1:

$$\tilde{\mathbf{x}}_t^{(i)} \sim \mathcal{N}(\mathbf{x}_{t-1}^{(i)}, k_t\mathbf{I}_2) \tag{6}$$

where a new particle of the prediction population is sampled from a normal distribution in which the mean is the i^{th} particle from the previous time step, and $k_t\mathbf{I}_2$ is a diagonal matrix driven by a diffusion factor to assure that the scatter scales in both dimensions are equal.

The accuracy of the system depends highly on the diffusion factor, k_t, as it determines the scatter of new particles. If the scatter is too high, the system tends to explore a large area of interest and this may result in high localization errors temporarily while exploring. If the scatter is too low, we restrict the system in a small neighborhood so that it cannot shoot the particles far enough, which prevents exploration of more accurate regions. We use two different diffusion factor strategies:

a: STATIC DIFFUSION

If the transmitter is assumed to move at a known constant velocity and if the RSSI measurements can discriminate the position accurately, an appropriate diffusion factor can be determined beforehand, by which the proposal density distributes the particles as far as required and the estimations are expected to be accurate:

$$k_t = k_{t-1} = k^* \tag{7}$$

We set the diffusion factor to a predetermined value, k^*, which ensures that the new particle predictions are scattered with respect to a distribution with a static covariance, or a static diffusion factor. A feasible value of this parameter is searched by running a filter with a discrete set of parameters. It is a problem that this factor may only be determined empirically, as we do not also know the velocity of the tracked object.

b: DECAYING DIFFUSION

The problem with the static diffusion factor may be overcome by reducing the diffusion factor gradually. As the observations are fed into the system, the estimations are expected to be more accurate. Beginning with a high diffusion factor, k_{max} for high exploration and decaying it gradually down to finer scales is another strategy. We set an initial diffusion factor,

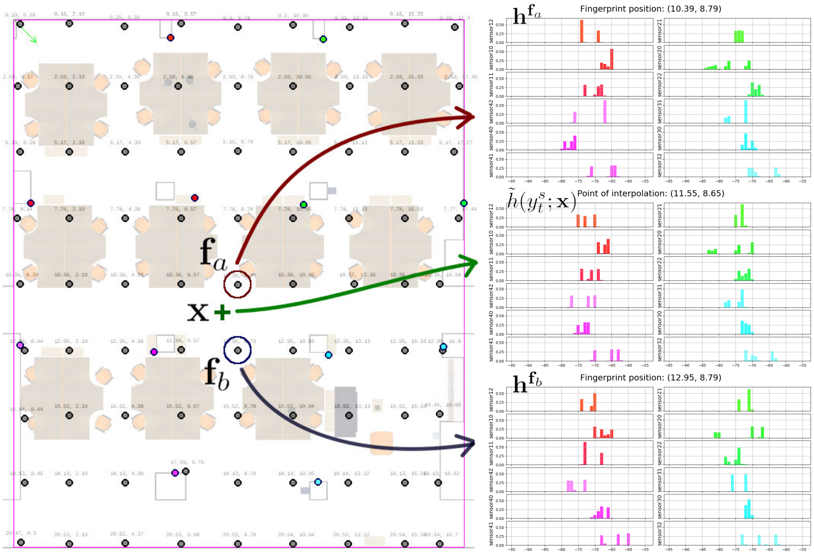

$\mathbf{h}^{\mathbf{f}_a}$
Fingerprint position: (10.39, 8.79)
$\tilde{h}(y_t^s;\mathbf{x})$
Point of interpolation: (11.55, 8.65)
$\mathbf{h}^{\mathbf{f}_b}$
Fingerprint position: (12.95, 8.79)
$\mathbf{f}_a$
$\mathbf{x}$
$\mathbf{f}_b$

FIGURE 2. Occupancy map: black cells denote the impassable regions (columns, furnitures or walls).

$k_0 = k_{max}$ and multiply it with a decay factor, η, at each step. The diffusion factor is reduced by some decay factor until it reaches a limit value, k_{min}.

$$\begin{aligned} k_0 &= k_{max} \\ k_t &= \begin{cases} \eta k_{t-1}, & \text{if } k_t > k_{min} \\ k_{t-1}, & \text{otherwise} \end{cases} \end{aligned} \tag{8}$$

The static diffusion strategy is a special case of the decaying diffusion strategy with $\eta = 1$ and $k_0 = k_{max}$.

3) PARTICLE EVALUATION

The particles are evaluated against a probabilistic radio map to determine their importance weights. A weight is quantified by the likelihood of being in that position with the measured RSSI value, however, we do not have such a map. Instead, we have access to some histograms on some reference points, i.e. the fingerprints. In this section, we first find an estimate for the probabilistic radio map that will emulate the emission density of the state space model, so that we can evaluate the particle predictions. Then, we update it with the architectural information as an occupancy map.

a: EVALUATION WITH PROBABILISTIC RADIO MAP ESTIMATE

To obtain the weight of a particle, we first need to estimate the RSSI distribution on the particle's specific position. We employ the Two-point AWC algorithm from Daniş and Cemgil [20]. The algorithm is used to estimate the corresponding RSSI histograms of a sensor on any position on the map using fingerprints. Note that this section is added to summarize the algorithm and to ensure the integrity of the whole methodology. The validity of the procedure and its relation to real measurements are detailed in the original article [20].

The histogram interpolation is a two-step process: Firstly, for a given position on the map, $\mathbf{x}$, we find the most appropriate pair of fingerprint positions from the set of all fingerprints, $(\mathbf{f}_a, \mathbf{f}_b) \in \mathcal{F}^2$. Secondly, we run AWC with the parameters, $\mathbf{x}$ as the target position and $(\mathbf{f}_a, \mathbf{f}_b)$ as the source positions. Briefly, the algorithm finds a transport plan of the probability measures on discrete positions by minimizing the cost of transferring measures from one histogram onto the other. The interpolation histogram on the specific position is then estimated by realizing this transport plan partially with respect to its distance to the fingerprint pair. An example interpolation is given in FIGURE 2.

We have a histogram on each of these positions for each sensor, $\mathbf{h}^{\mathbf{f}_a}$ and $\mathbf{h}^{\mathbf{f}_b}$, and we are to estimate the histograms of RSSI measurements on $\mathbf{x}$, $\tilde{h}(y_t^s; \tilde{\mathbf{x}}_t, \alpha, \beta)$. The algorithm specific parameters, α is a similarity measure that is tuned with respect to the distance of the target position to each of the source positions, and β controls if the interpolation is a linear interpolation or a displacement interpolation.

With an estimate of the probabilistic radio map defined on any point on the map, we evaluate the particles using the measured RSSI value to obtain their importance weights.

$$\begin{aligned} \omega_t^{(i)} &= p(y_t^s | \tilde{\mathbf{x}}_t^{(i)}) \\ &= \tilde{h}(y_t^s; \tilde{\mathbf{x}}_t, \alpha, \beta) \end{aligned} \tag{9}$$

b: EVALUATION WITH OCCUPANCY MAP

Any area that the system is installed in may have impassable (occluded) regions, such as building columns, walls, cabinets or tables. The transmitter is not expected to pass through these occluded regions. These impassable regions of the area are labeled beforehand, so that the particles that get generated on these regions are evaluated accordingly.

The occupancy grid is denoted by $r(\mathbf{x})$:

$$r(\mathbf{x}) = \begin{cases} 0, & \text{if } \mathbf{x} \text{ is in the occupied area,} \\ 1, & \text{otherwise} \end{cases} \tag{10}$$

For particle evaluation with occlusion, we have two choices: (i) these particles are either resampled until they fall in the passable region, or (ii) the weights of such particles are directly set to zero. The former method (i), would be the desired action in order to keep the efficient sample size intact, but this action may lead to several resampling attempts especially if the particles are at the border, resulting in a drastic slow down of the current step. We use the latter method (ii) to keep the evaluation duration stable by setting the particle weights to zero if they are out of the passable area, which means that the particles are sterilized not to produce offsprings in the resampling and nor are they taken into consideration for weighted mean estimation:

$$\omega_t^{(i)} = \tilde{h}(y_t^s; \tilde{\mathbf{x}}_t^{(i)}, \alpha, \beta) r(\tilde{\mathbf{x}}_t^{(i)}) \tag{11}$$

where the importance weight of the i^{th} particle, $\omega_t^{(i)}$, is also multiplied with the corresponding occupancy information $r(\tilde{\mathbf{x}}_t^{(i)})$, which sets the weight to zero if the sampled position, $\tilde{\mathbf{x}}_t^{(i)}$, is in an occupied zone, or leaves untouched otherwise. A sample occupancy map is given in FIGURE 3.

The importance weights are normalized before the resampling stage:

$$\tilde{\omega}_t^{(i)} = \frac{\omega_t^{(i)}}{\sum_{i=0}^{N} \omega_t^{(i)}} \tag{12}$$

4) RESAMPLING

A resampling strategy is used to overcome the degeneracy problem of the algorithm, that is to avoid a situation where all importance weights but one are zero. We use the systematic resampling strategy [38], whose time complexity is of order $O(N)$, more efficient than the multinomial or stratified resampling techniques [39]. We also add an evaluation of the three traditional resampling techniques for this setup in the Section IV.

$$\mathbf{x}_t^{(i)} \sim Syst(\tilde{\mathbf{X}}_t, \tilde{\mathbf{w}}_t), \quad i \in [1..N] \tag{13}$$

The SMC filter for BLE Localization is summarized in Algorithm 1.

C. ADAPTIVE SMC FILTER WITH PARAMETER SAMPLING FOR BLE LOCALIZATION

In this section, we describe the adaptive SMC filter that jointly estimates the position and the unknown variable diffusion factors. Compatible with the proposed hidden Markov model (FIGURE 1), we assume to have two layers of latent variables, one for the position and the other for the diffusion factors.

We modify the generic SMC filter defined in Section II-B by first adding the diffusion particles. We will denote the diffusion (factor) particles with the superscripted version of the diffusion factor, $k_t^{(i)}$. The position particles are also modified to have two indices in superscript, $\mathbf{x}_t^{(i,j)}$, with i for the diffusion particle it depends and j for the index in the set of position particles that depend on the same diffusion particle. We also introduce the similar versions for the position particle weights, $\omega_t^{(i,j)}$, and diffusion particle weights, $\upsilon_t^{(i)}$.

The generic SMC filter is also modified in terms of the number of particles. We assume that multiple position

Algorithm 1 SMC Filter for BLE Localization

```
1: Initialization
   Sample a set of particles:
      for $i \in [1..N]$, $\mathbf{x}_0^{(i)} \sim \mathcal{U}(\mathcal{A})$
   Initialize diffusion factor:
      $k_0 = k_{max}$
2: Importance Sampling
   Update the time index
      $t \leftarrow t + 1$
   Sample a new set of particles from the current set:
      for $i \in [1..N]$, $\tilde{\mathbf{x}}_t^{(i)} \sim \mathcal{N}(\mathbf{x}_{t-1}^{(i)}, k_t \mathbf{I}_2)$
   Evaluate the proposed set with the radio frequency and occupancy maps to obtain the importance weights:
      for $i \in [1..N]$, $\omega_t^{(i)} = \tilde{h}_{y_t^s}(\tilde{\mathbf{x}}_t^{(i)}, \alpha, \beta) r(\tilde{\mathbf{x}}_t^{(i)})$
   Normalize the importance weights:
      for $i \in [1..N]$, $\tilde{\omega}_t^{(i)} = \frac{\omega_t^{(i)}}{\sum_{i=0}^{N} \omega_t^{(i)}}$
   Update the diffusion factor:
      $k_t = \eta k_{t-1}$
3: Resampling
   Resample $N$ particles according to the importance weights
      for $i \in [1..N]$, $\mathbf{x}_t^{(i)} \sim Syst(\tilde{\mathbf{X}}_t, \tilde{\mathbf{w}}_t)$
4: Recursion
   Go back to 2, if not at the end of the trajectory
```

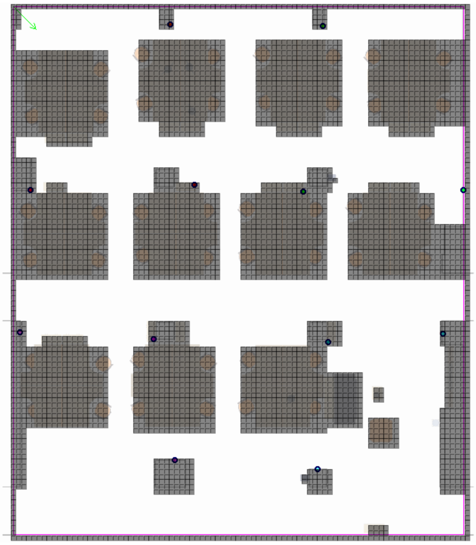


FIGURE 3. Occupancy map: black cells denote the impassable regions (columns, furnitures or walls).

particles can be instantiated by a diffusion particle. We will denote the size of the diffusion particles with D and the size of the position particles generated by each diffusion particle with P. We can also make the algorithm only one chain dependent by setting the population size of the other chain to 1.

1) INITIALIZATION

We do the initial sampling of the diffusion particles, $k_0^{(i)}$, that are distributed along a uniform distribution. The position particles are initially sampled like in the original SMC filter for localization, but for a different sample size of P per diffusion particle.

$$k_0^{(i)} \sim \mathcal{U}([k_{min}, k_{max}]),\ \ i \in [1..D]$$

$$\mathbf{x}_0^{(i,j)} \sim \mathcal{U}(\mathcal{A}),\ \ i \in [1..D],\ \ j \in [1..P] \tag{14}$$

2) IMPORTANCE SAMPLING

Diffusion particles are sampled along a gamma distribution, aligned with the transition density given in (3) with $i \in [1..D]$. Position particles are sampled as in Section II-B2, but for two indices: $i \in [1..D]$ for the diffusion particle and $j \in [1..P]$ defining the index in the population sampled by the associated diffusion particle:

$$\tilde{k}_t^{(i)} \sim \mathcal{G}(a_t^{(i)}, b_t^{(i)}),\ \ i \in [1..D]$$

$$\text{with } a_t^{(i)} = \frac{(k_{t-1}^{(i)})^2}{\nu} + 1$$

$$\text{and } b_t^{(i)} = \frac{\nu}{k_{t-1}^{(i)}}$$

$$\tilde{\mathbf{x}}_t^{(i,j)} \sim \mathcal{N}(\mathbf{x}_t^{(i,j)}, k_t^{(i)}\mathbf{I}_2)),\quad i \in [1..D],\ j \in [1..P] \tag{15}$$

Small values of the sensitivity make the diffusion estimations rigid, keeping the weighted average of the diffusion factor almost static, whereas with the higher values we observe rapidly varying values for the same statistic, making the diffusion factor estimations behave aligned with the measurement errors. As a consequence, higher values may seem to be preferred, but we will show that unnecessarily high sensitivity values can also cause undesired fluctuations for the diffusion factor estimations.

3) PARTICLE EVALUATION

The position particles, $\omega_t^{(i,j)}$, are evaluated against the frequency map and the occupancy map to obtain the particle weights as before (see Section II-B3). The diffusion particles, $\upsilon_t^{(i)}$, are evaluated by accumulating the corresponding position particle weights.

$$\begin{aligned} \omega_t^{(i,j)} &= \tilde{h}(y_t^s; \tilde{\mathbf{x}}_t^{(i,j)}, \alpha, \beta) r(\tilde{\mathbf{x}}_t^{(i,j)}) \\ \upsilon_t^{(i)} &= \sum_j \omega_t^{(i,j)} \end{aligned} \tag{16}$$

Algorithm 2 Parameter Sampling Based Adaptive SMC Filter for BLE Localization

```
1: Initialization
   Initialize a set of particles:
     for $i \in [1..D]$, $k_0^{(i)} \sim \mathcal{U}([k_{min}, k_{max}])$
       for $j \in [1..P]$, $\mathbf{x}_0^{(i,j)} \sim \mathcal{U}(\mathcal{A})$
2: Importance Sampling
   Update the time index
     $t \leftarrow t+1$
   Sample a new set of particles from the current sets:
     for $i \in [1..D]$, $\tilde{k}_t^{(i)} \sim \mathcal{G}(a_t^{(i)}, b_t^{(i)})$,
       for $j \in [1..P]$, $\tilde{\mathbf{x}}_t^{(i,j)} \sim \mathcal{N}(\mathbf{x}_{t-1}^{(i,j)}, k_t^{(i)}\mathbf{I}_2)$
   Evaluate the proposed set with the radio frequency and occupancy maps to obtain the importance weights:
     for $i \in [1..D]$,
       for $j \in [1..P]$, $\omega_t^{(i,j)} = \tilde{h}_{y_t^s}(\tilde{\mathbf{x}}_t^{(i,j)}, \alpha, \beta) r(\tilde{\mathbf{x}}_t^{(i,j)})$
       $\upsilon_t^{(i)} = \sum_j \omega_t^{(i,j)}$
   Normalize the importance weights:
     $\tilde{\omega}_t^{(i,j)} = \frac{\omega_t^{(i,j)}}{\sum_{i,j} \omega_t^{(i,j)}}$
     $\tilde{\upsilon}_t^{(i)} = \frac{\upsilon_t^{(i)}}{\sum_i \upsilon_t^{(j)}}$
3: Resampling
   Resample $D \times P$ particles according to the importance weights
     for $i \in [1..D]$, $k_t^{(i)} \sim Syst(\tilde{\mathbf{k}}_t, \tilde{\boldsymbol{\upsilon}}_t)$
       for $j \in [1..P]$, $\mathbf{x}_t^{(i,j)} \sim Syst(\tilde{\mathbf{X}}_t^{(i)}, \tilde{\mathbf{w}}_t)$
4: Recursion
   Go back to 2, if not at the end of the trajectory
```

Both set of particles are then normalized:

$$\begin{aligned} \tilde{\omega}_t^{(i,j)} &= \frac{\omega_t^{(i,j)}}{\sum_{i,j} \omega_t^{(i,j)}} \\ \tilde{\upsilon}_t^{(i)} &= \frac{\upsilon_t^{(i)}}{\sum_i \upsilon_t^{(j)}} \end{aligned} \tag{17}$$

4) RESAMPLING

With the normalized weights for the set of particles, the resampling stage is the same with the resampling in Section II-B4. The SMC filter with parameter sampling is given in Algorithm 2. The bold vectors or matrices, $\tilde{\mathbf{k}}_t$, $\tilde{\boldsymbol{\upsilon}}_t$, $\mathbf{X}_t^{(i)}$ and $\tilde{\mathbf{w}}_t$, stand for the sets of the corresponding particle variables.

III. TEST SETUP

A. TEST ENVIRONMENT AND COMPONENTS

We install our equipment in an open office area of size 17 × 20 m^2 to test our work. The plan of the area is given in FIGURE 4. In our scenario, we assume that an emitter (BLE beacon) is navigating in the area and we expect to accurately track its position in the area coordinate system.

The office environment is decorated with 12 sensors (Bluetooth adapters) that capture the BLE packets emitted by the

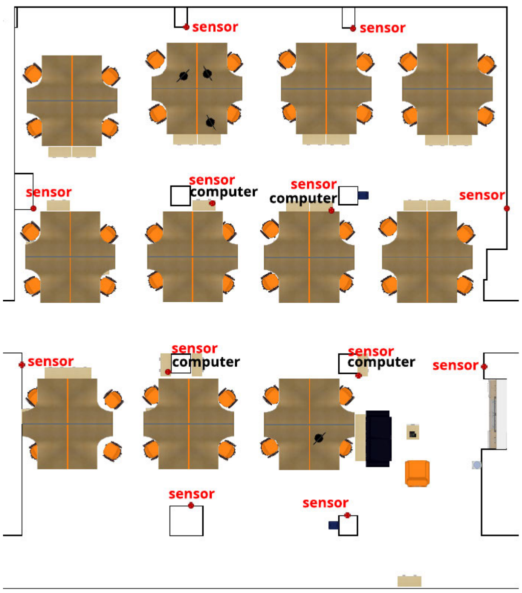


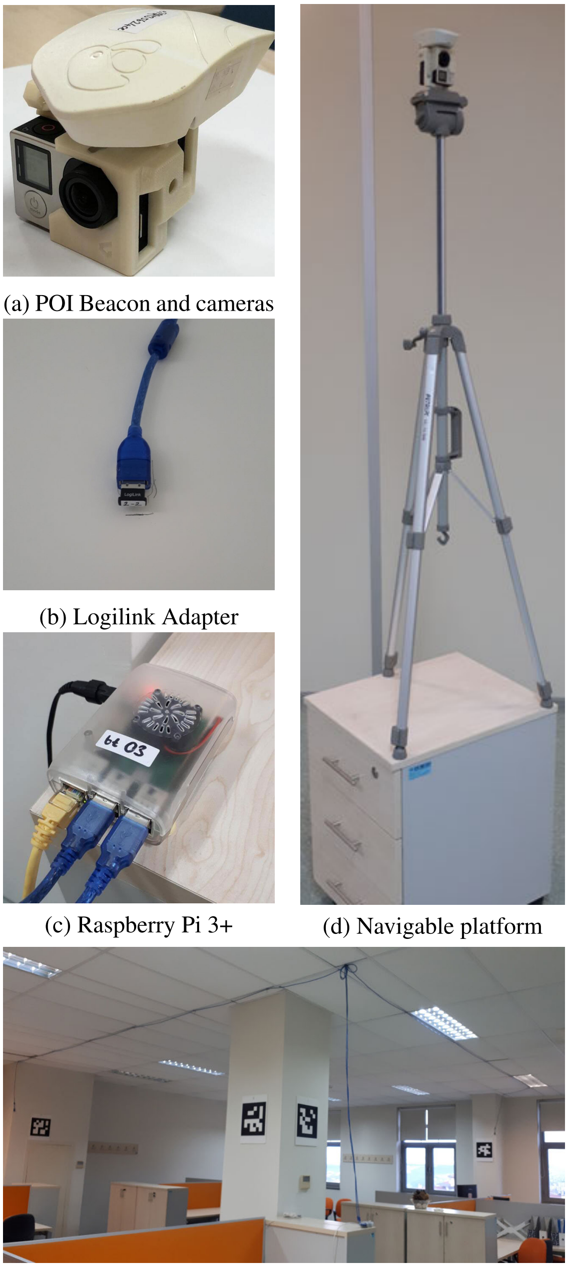
FIGURE 4. Plan of the test area, with sensors and computers marked as red circles and black squares respectively.

BLE beacon. The sensors are administered by four computers. We use off-the-shelf Raspberry Pi 3+ single board computers as sensing and publishing units (FIGURE 5c). These computers are connected to a local area network through Ethernet. All of the computers are running Raspbian Stretch operating system with Robot Operating System (ROS) Melodic middleware [40].

As BLE packet transmitters, we use POI Beacons, based on Nordic Semiconductor, running on 32bit ARM Cortec with 256KB flash and 16KB RAM, provided by POI Labs[1](see FIGURE 5a). The beacons are capable of 2.4Ghz Bluetooth 4.0 BLE protocol and have a signal range of 70 meters. They are programmed to publish 2 BLE identity packets per second. Powered by 2 AA batteries, they can be functional up to 4 years with the current configuration.

Compatible with the BLE protocol, Logilink Bluetooth 4.0 adapters are used to capture BLE packets (see FIGURE 5b). To expand the coverage, we attach two of these adapters to each processor with USB extension cables (see FIGURE 5e). We also employ the onboard Bluetooth modules of the computers connected through UART bus, attaining a total of 12 Bluetooth sensors.

For data collection, we designed an affordable navigable platform, (see FIGURE 5d), which is composed of a wheeled cabinet, a tripod and a custom case for the cameras. Two cameras and a BLE beacon are bundled into the custom case (see FIGURE 5) that is raised by a tripod to gain a clear view of the surrounding area. The tripod is attached on a mobile cabinet to collect data while navigating. The cameras are used to estimate the ground truth pose data very accurately. The ground truth is collected using a special decoration of AR tags in the environment.



(a) POI Beacon and cameras

(b) Logilink Adapter

(c) Raspberry Pi 3+

(d) Navigable platform

(e) View from the area and cabling of Bluetooth adapters

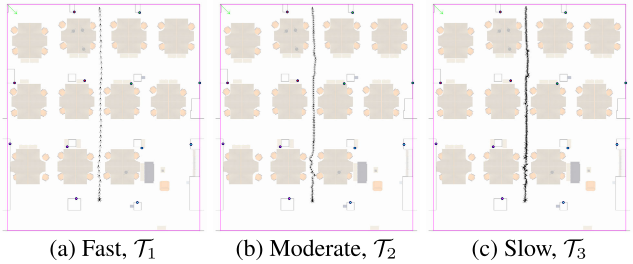
FIGURE 5. Test components.

B. DATA COLLECTION

We gather stationary RSSI measurements that will build into fingerprints. The platform is placed on a position and the data collection system is triggered. The system dumps the RSSI data along with the sensor and beacon addresses published by the sensors for 30 minutes and this procedure is repeated for 81 different positions on the map. Finally we have normalized histograms of RSSI measurements for each sensor on several points (see FIGURE 2).

We create an occupancy map for impassable regions. The walls, tables, columns, closets and couches are considered to be impassable areas on the map. We determine a grid matrix

[1] poilabs.com



(a) Fast, $\mathcal{T}_1$ (b) Moderate, $\mathcal{T}_2$ (c) Slow, $\mathcal{T}_3$

FIGURE 6. **Synthetic test trajectories.**

with a width size of 0.2 meters, set the regions covered by such objects occupied areas, and label these areas manually (see FIGURE 3). The labels for the occupied grid positions are set to zero, and the rest are set to one.

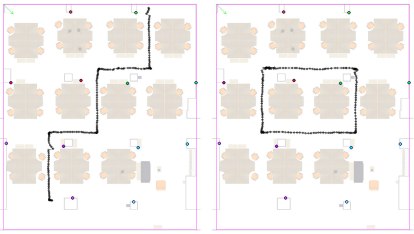
To avoid the on-the-fly computation of histogram interpolations on arbitrary map positions, we generate the probabilistic radio map and the occupancy map as grid maps with a resolution of 0.2 meters, before running the actual experiments. This prior preparation makes the estimation of occupancy map or RSSI histograms merely a memory based constant time lookup operation in runtime if position indexing is properly provided.

Finally, using the navigable platform, we capture RSSI measurements, the corresponding beacon and sensor addresses, timestamp of the capture and the ground truth position of the platform while navigating it on predetermined trajectories. The platform is navigated by hand. The vision-based ground truth data collection system is activated, so each RSSI measurement is labeled with its ground truth. We determine three trajectories with accompanying RSSI values: $\mathcal{T}_1$, a trajectory at a fast pace ($\sim$ 0.9m/s); $\mathcal{T}_2$, one at a moderate pace ($\sim$ 0.4m/s); and $\mathcal{T}_3$, another with a slow pace ($\sim$ 0.1m/s). Since the algorithms do not take the shape of the trajectories into consideration but the continuity, we use linear trajectories to keep the paces steady and controlled. These trajectories are shown in FIGURE 6. On these trajectories we synthesize two experimental sets of RSSI data: (i) one set of very accurately synthesized RSSI measurements using the grid structure of the frequency maps ($\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_3$), (ii) and another one of highly noisily synthesized RSSI data using the closest fingerprint information ($\mathcal{T}_1'$, $\mathcal{T}_2'$ and $\mathcal{T}_3'$).

We perform the final batch experiments on the real RSSI data captured while navigating in the environment. The navigable platform is driven at varying paces and the trajectory is not linear. We use two different trajectories with associated real RSSI data: $\mathcal{T}_4$, a zigzagging trajectory; and $\mathcal{T}_5$, a rectangular trajectory (see FIGURE 7).

IV. EXPERIMENTS AND RESULTS

The experiments are performed on an Intel Xeon E5-2697A running at 2.60GHz. For the experiments, we run the SMC filters of different models with different configurations. A sample visualization of a running SMC filter is given in FIGURE 8.



(a) Zigzagging, $\mathcal{T}_4$ (b) Rectangular, $\mathcal{T}_5$

FIGURE 7. **Real test trajectories.**

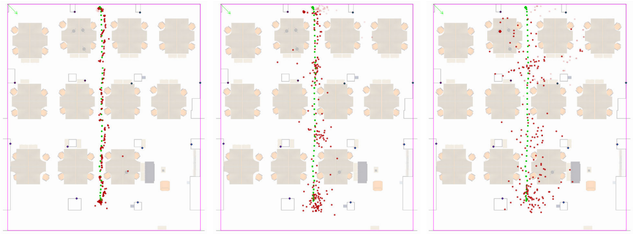


FIGURE 8. **Visualizations of SMC filter experiment samples with different diffusion factors. Green dots are the ground truth positions, red dots are the estimated positions.**

To statistically assess the results, each configuration is run for 50 times in parallel and repeated for different trajectories. We vary the following parameters in the following sets for different models:

- Static diffusion model
  - static diffusion factor: $k \in [0.001, 2.0]$
  - particle population size: $N \in \{1K, 2K, 5K, 10K\}$
- Decaying diffusion model
  - decaying diffusion factor: $\eta \in \{0.9, 0.95, 0, 99\}$
  - diffusion factor higher limit: $k_{max} = 5.0$
  - diffusion factor lower limit: $k_{min} \in [0.01, 2.0]$
  - particle population size: $N = 1K$
- Adaptive diffusion model
  - initial diffusion particles: $k_0 \in [0.00001, 5.0]$
  - sensitivity: $\nu \in [10^{-8}, 10^{-2}]$
  - particle population sizes: $(D, P) \in \{(500, 1), (333, 2), (250, 3), (200, 4), (100, 9), (50, 19), (40, 14), (20, 49), (10, 99), (5, 199), (4, 249), (2, 499), (1, 999)\}$

We evaluate each experiment by the accuracy of the estimated trajectory. We measure this accuracy by the mean and median of the Euclidean distances between the weighted position means, or estimations, and their corresponding ground truth positions. In FIGURE 9, we show a sample of trajectory errors throughout an experiment, the distribution and cumulative distribution of these errors. The very first estimations of a trajectory yield high errors naturally, and as more data are fed, the estimations tend to give lower errors. We rather rely on the medians of errors than the means, because the errors obtained after an experiment form right skewed distributions. This skewness is mostly due to the temporary far divergences of the particle clouds from the true positions.

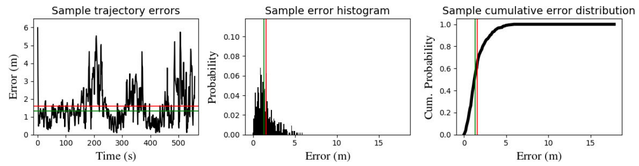


FIGURE 9. Sample error distributions. From left to right: Trajectory errors throughout an experiment, histogram of the corresponding errors, and cumulative distribution of the errors. The green and red lines indicate the median and mean of the distribution respectively.

We perform three types of main experiments: The first set of experiments aims to find the best strategy along with the most appropriate parameters for the SMC algorithm on synthetically and very accurately sampled trajectories. The first set is also for setting a proof of concept for the proposed methods and determining the parameters that are to be used in the experiments that follow. For the second set of experiments, we generate noisy RSSI data using the closest fingerprint information. This second set is used for finding the generalized parameters for the different models of SMC filters. The third set of experiments evaluate the performance of the SMC filter on the trajectories with real RSSI measurements collected by navigating the test setup in the environment. Narrowing down the parameter set in the first and second, we compare and focus on the model parameters that can be used in real world conditions where the data is degenerated due to various factors.

A. PARTICLE POPULATION SIZE SELECTION

According to the nature of the SMC algorithms, with larger particle population sizes the algorithms approximate better, but higher particle sizes do not suit the real world conditions because of high processing power requirements. Before elaborating the error comparisons, we first select a reasonable particle population size by comparing the effect on the experiment run times with the frequency of the captured. In FIGURE 10, we show the step processing durations throughout the experiments and corresponding trajectory error distributions. For the experimental trajectories $\mathcal{T}_3$ and $\mathcal{T}_3'$, we set safe static diffusion factors of 0.2 and 0.3 respectively. Safe parameters tend to explore sufficiently and does not get stalled. We see that varying the population size from 1K to 10K does not significantly reduce the error for the specific experiments. Hence, we choose a safe particle population size of 1K particles, which allows the system to process about 25 data points per second. With the current setup, if the processing were to be performed in real time, such a configuration would rarely miss any captured data.

Since there exist two layers that sample particles in the adaptive strategy, a comparable particle population size can be computed with $N_{ada} = D + DP$, because after D diffusion particles are sampled, for each sampled diffusion particle P position particles are sampled. The following sections use compatible population sizes of diffusion particles and position particles.

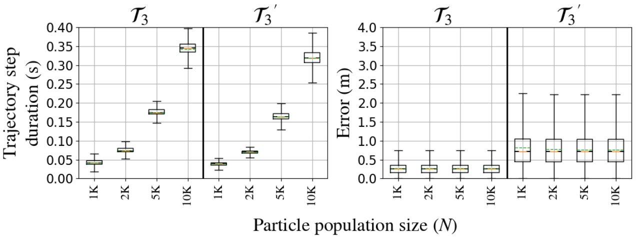


FIGURE 10. Duration distributions (left) and corresponding error distributions (right) with respect to different particle population sizes.

B. RESAMPLING STRATEGY SELECTION

Resampling strategy is also very crucial in SMC. We compare three traditional resampling strategies: Multinomial, systematic and stratified. Whereas multinomial resampling strategy is known to approximate better as it simulates the real particle weight distribution, it has a higher computational complexity compared to its counterparts. We repeat the experiments for different resampling strategies with the same static diffusion factors, 0.2 and 0.3, for the trajectories, $\mathcal{T}_3$ and $\mathcal{T}_3'$. In FIGURE 11, we compare three traditional resampling techniques with respect to their running times and error distributions.

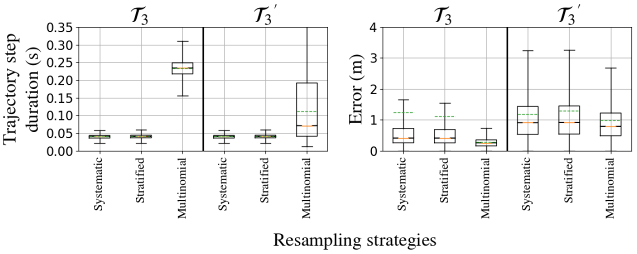


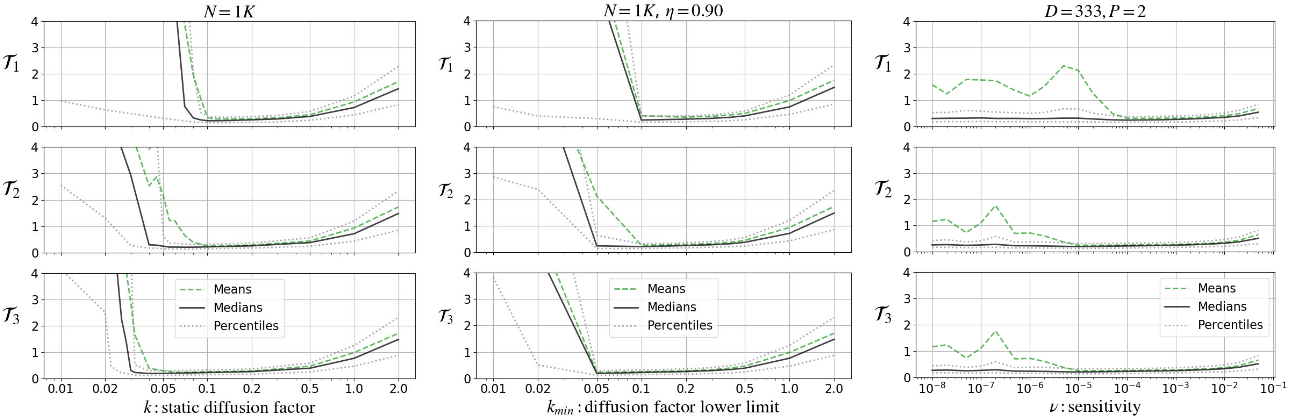
FIGURE 11. Duration distributions (left) and corresponding error distributions (right) with respect to different resampling strategies sizes.

The results show that varying the resampling strategy does not significantly improve the error statistics, but running times differ considerably. Employing a multinomial resampling strategy boosts the running times, however, corresponding error distributions have lower variances. Nonetheless, other strategies are preferable for their lower running times. There is no significant difference between stratified and systematic resampling strategies. Because the systematic resampling is also known to be the best with respect to the computational complexity and random numbers used [39], we prefer it in the following experiments.

C. EXPERIMENT I: ACCURATELY SYNTHESIZED RSSI DATA

For an exhaustive parameter search, we regenerate synthetic RSSI data on the collected trajectory positions. We aim both to show a proof of concept of an efficient adaptive parameter sampling method for the SMC filter in BLE based indoor localization, and find the best filter parameters for the given trajectories.

Among the three different diffusion factor strategies, the static diffusion factor performs better than others as



(a) Static diffusion strategy

(b) Decaying diffusion strategy

(c) Adaptive diffusion strategy

FIGURE 12. Accurately synthesized data: Error (m) means and medians for diffusion factor strategies. The black, green and gray curves represent the median, mean and percentiles (25% and 75%) respectively.

expectedly if an optimal diffusion factor is given beforehand. We report the error means and medians with respect to both different static diffusion factors, keeping the population size at 1K. As shown in FIGURE 12a, we achieve the lowest medians around 0.1, 0.07 and 0.04, compatible with the paces of the trajectories. Moreover, the trajectory estimates are very accurate (best errors around 0.22 meters) and thus the method is very appropriate as the RSSI data are synthesized using the radio frequency map grids of size 0.2 meters.

These results show that in order to achieve accurate estimations, we should set the correct values for the static diffusion factors before the actual filter is run, but these correct values cannot be known beforehand as velocities or observation noises are unknown, or moreover the velocity can vary throughout an experiment. For the paces comparable with the walk pace of a person, setting a value between 0.1 and 0.5 would be the best for the static diffusion factor, but for the values just under the best factor, we risk of falling into a stalled state where the estimations cannot keep up with the actual trajectory, because of the lack of exploration tendency. This strategy shows that a high static diffusion factor should be chosen to cope with the varying and unknown velocities, however, we see that for higher values the SMC filter loses the focus and tends to explore with largely scattered particle clouds.

Selecting a decaying diffusion factor that behaves exploratorily in the beginning and more exploitatively through the end, shows that we reach a more generalized system for some of the experiments, but we still require to select a covariance factor lower limit where the decaying should not surpass. With higher decay rates we attain more generalized error distributions, but with lower decay rates lower errors are obtained. Setting appropriate lower limits makes the system no different than selecting static diffusion factors (see FIGURE 12b).

The proposed approach, the adaptive diffusion factor, however, yields more generalized results by avoiding falling into a stalled state. Sampling the diffusion factor helps the system maintain a generalized behavior, that is, when the error gets higher exploration is favored, and exploitative behavior otherwise. FIGURE 13 shows a gist of this behavior on a sample run. The temporary peaks of the weighted mean diffusion factor can be seen when the positioning errors are boosted. The diffusion factor adapts itself to the changing conditions (i.e. loss of particle focus).

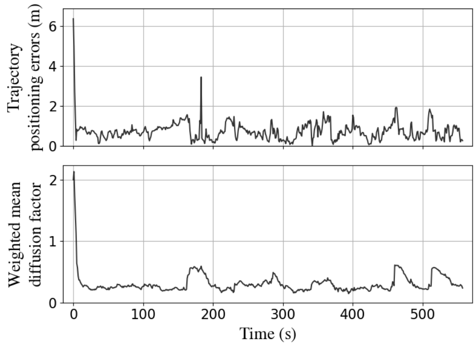


FIGURE 13. Adaptive behavior of a sampled diffusion factor. The weighted average of the diffusion factor accompanies the error fluctuations of the position estimates.

The diffusion factor is not set too low or too high, both of which would result with high positioning errors. In FIGURE 12c, the error mean and medians are plotted against different sensitivity values. We see that even varying the sensitivity dramatically does not boost the errors like in the static or decaying cases. The medians of the errors tend to stay around a constant value, which is to say that we have a more generalized algorithm that can tune its diffusion factor, k, according to the requirements. Selecting a sensitivity value about 2×10^{-3} is appropriate for our algorithm to achieve low, but more importantly consistent error distributions.

For each set of experiments we also show the lowest errors achieved and the corresponding parameters. For the synthetic data, these statistics are given in TABLE 1. According to the

TABLE 1. Best error statistics (m) for accurately and noisily synthesized RSSI measurements on different trajectories.

		Static	Decaying	Adaptive
		$N = 1K$	$N = 1K$	$D = 333$ $P = 2$
$\mathcal{T}_1$	Par.	$k^* = 0.1$	$k_{min} = 0.2$ $\eta = 0.90$	$\nu = 1.0 \times 10^{-4}$
	Median	**0.227**	0.239	0.235
	Mean	0.356	0.349	**0.312**
$\mathcal{T}_2$	Par.	$k^* = 0.07$	$k_{min} = 0.1$ $\eta = 0.90$	$\nu = 2.0 \times 10^{-5}$
	Median	0.202	0.217	**0.198**
	Mean	0.248	0.267	**0.247**
$\mathcal{T}_3$	Par.	$k^* = 0.04$	$k_{min} = 0.05$ $\eta = 0.90$	$\nu = 1.0 \times 10^{-5}$
	Median	**0.183**	0.185	0.187
	Mean	**0.222**	0.230	0.229
$\mathcal{T}'_1$	Par.	$k^* = 0.4$	$k_{min} = 0.5$ $\eta = 0.90$	$\nu = 2.0 \times 10^{-4}$
	Median	0.748	0.777	**0.675**
	Mean	**0.780**	0.836	0.803
$\mathcal{T}'_2$	Par.	$k^* = 0.3$	$k_{min} = 0.2$ $\eta = 0.90$	$\nu = 5.0 \times 10^{-4}$
	Median	0.671	0.675	**0.654**
	Mean	0.802	0.830	**0.734**
$\mathcal{T}'_3$	Par.	$k^* = 0.03$	$k_{min} = 0.2$ $\eta = 0.90$	$\nu = 1.0 \times 10^{-4}$
	Median	0.717	0.737	**0.688**
	Mean	0.812	0.866	**0.763**

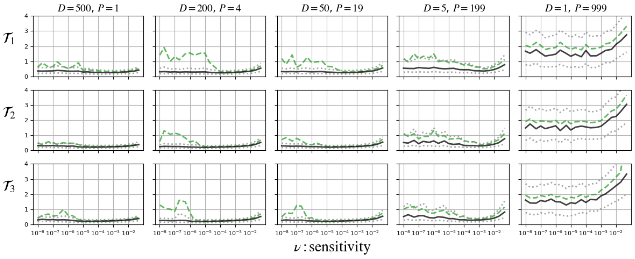


FIGURE 14. Error distributions for different sensitivity values and different particle size configurations.

tested parameters, the static diffusion strategy performs the best. This is an expected behavior, since we note that these experiments have constant velocity. Besides, an adaptive diffusion strategy shows comparable performances with the two other strategies.

D. VARYING THE DIFFUSION PARTICLE POPULATION SIZE

We set the total sampled particle size to 1K, but in order to decide how many samplings each layer should perform, we examine the error distributions with varying diffusion and position particle sizes. FIGURE 14 summarizes the error distributions with respect to the sensitivity. We see that by distributing the particle sizes to the layers, we reach lower error rates for specific sensitivity values ($D = 200, P = 49$), but narrower thus highly stable plots are obtained with increased diffusion particle size ($D = 500, P = 1$).

To determine an appropriate particle size tuple, we look finally at the timings of each option. FIGURE 15 compares the step durations and errors of static and decaying diffusion model at 1K with the sampling models of various configurations. The adaptive model with a diffusion particle size of 500 has the lowest run times with error rates comparable to the opposing configurations. This is due to the fact that sampling a position particle is more time consuming than sampling a diffusion particle. The parameters $(D, P) \in \{(500, 1), (333, 2), (250, 3)\}$ for the adaptive strategy are favorable, because they have lower run times and highly stable error rates.

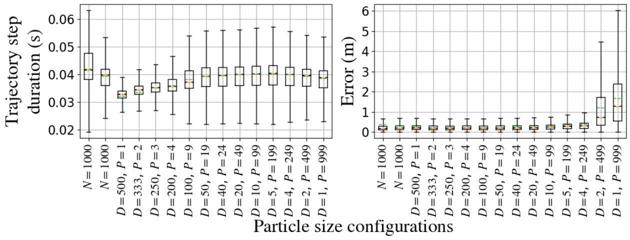


FIGURE 15. Duration (left) and error rate (right) comparison of the different adaptive strategy configurations with the static and decaying models, the first and second distributions respectively from the left.

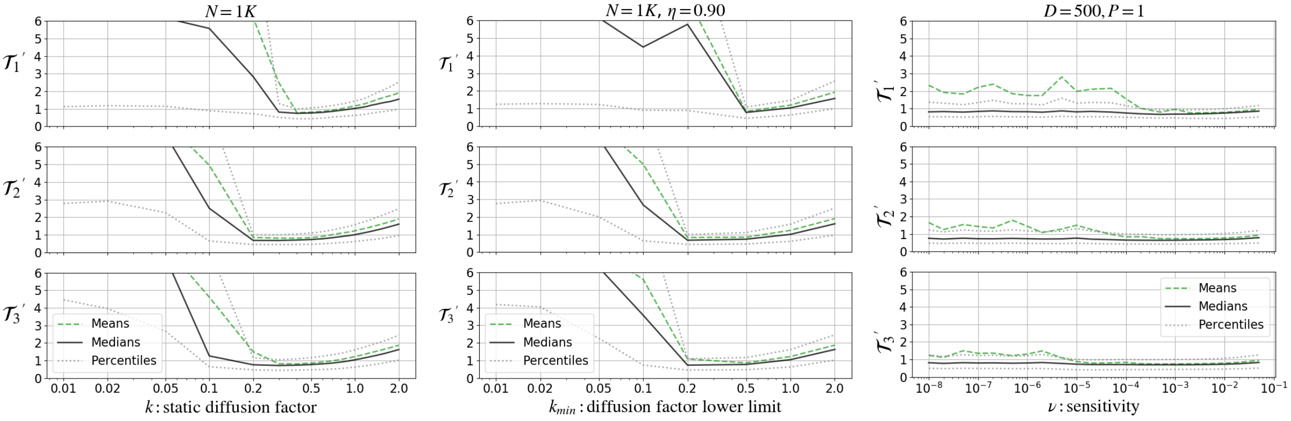
E. EXPERIMENT II: NOISILY SYNTHESIZED RSSI DATA

The measurement model of the first set of experiments is designed to be very accurate, and does not project to real life conditions. For the second set of experiments, we disrupt the generation model to synthesize highly noisy RSSI data for the trajectories. We mark the identifiers of the trajectories with the primed versions of the same trajectories in the first set. ($\mathcal{T}'_1$, $\mathcal{T}'_2$ and $\mathcal{T}'_3$). The aim of this set is to show whether the findings after the first set is still valid for a more generic setup.

These data are synthesized using the fingerprints. The RSSI data point on a trajectory position is resampled upon the closest corresponding fingerprint histogram, which makes the generated RSSI data highly noisy with respect to the previous data set. Considering the gaps between the fingerprint positions, estimation errors up to 1 m is acceptable.

FIGURE 16 displays a similar behavior for the strategies, but with higher error rates. For the static and decaying diffusion, we obtain U-shaped graphics, left of which points the space of stall, and the right shows the space of exploratory behavior. The space of the adaptive strategy errors is far from a U-shape. Though, if the search space is enlarged, we surely experience some deteriorated error distributions, but with a large enough search space we obtain stable error distributions for the adaptive diffusion strategy. This is interpreted as the diffusion rate is automatically tuned for better estimations wherever it was initialized.

The adaptive strategy performs also better when the best parameters are compared. We observe in TABLE 1 that for all of the experiments the adaptive strategy reaches the lowest error medians and means. We conclude that, if the measurements are noisy, an adaptive strategy based on parameter estimation by sampling leads to the better estimations.



(a) Static diffusion strategy (b) Decaying diffusion strategy (c) Adaptive diffusion strategy

FIGURE 16. Noisily synthesized RSSI data: Error (m) means and medians for diffusion factor strategies. The black, green and gray curves represent the median, mean and percentiles (25% and 75%) respectively.

F. EXPERIMENT III: REAL RSSI DATA

For the real data, we use two different trajectories whose RSSI measurements are captured with the navigable platform detailed in Section III. The RSSI data are collected simultaneously with the ground truth positions. The position estimations by the SMC filters are evaluated with these ground truth positions. The nature of the RF signal and the lack of a dead reckoning information prevent the system from making accurate estimations. Different than the first two set of experiments, here the emission density is very noisy and the navigation velocity is variable.

Even with the handicapped conditions, FIGURE 17 shows that the adaptive diffusion strategy keeps its steadiness for a large range of sensitivity values. We remark also that, for the real data set, we choose particle parameter sizes as $(D, P) = (50, 19)$, that is, at each iteration we sample 50 particles for diffusion factor and 19 position particles for each diffusion factor. Even though the best parameters are found empirically, we can generalize that choosing a larger size for the diffusion particle population produces more accurate estimations (see Section IV-D).

We compare finally the best parameter configurations for each strategy. The adaptive diffusion strategy outperforms the other strategies with respect to stability and error performance. FIGURE 17 shows that the static and decaying strategies are prone to fall into a stalled state if the diffusion parameters are selected to be too low, but the adaptive diffusion strategy does not get stalled for a comparatively large space of sensitivity values. Moreover, TABLE 2 shows that the adaptive diffusion strategy has the best median and mean error rates.

We report the best sets of parameters that are found empirically. It can be seen from the results that the best set of parameters may vary even for different trajectories in the same area. One may think that the adaptive strategy perform better for the chosen specific parameters, but the performance of the algorithm is similar for a large range of its sensitivity values. We can generalize that selecting an arbitrary sensitivity value from the ranges will perform acceptably without ever knowing these empirically best values.

TABLE 2. Best error statistics (m) for the experiments that use real RSSI measurements.

		Static	Decaying	Adaptive
		$N = 1K$	$N = 1K$	$D = 50$ $P = 19$
$\mathcal{T}_4$	Par.	$k^* = 1.1$	$k_{min} = 1.1$ $\eta = 0.90$	$\nu = 5.0 \times 10^{-4}$
	Median	2.434	2.45	**2.223**
	Mean	3.362	3.284	**3.064**
$\mathcal{T}_5$	Par.	$k^* = 1.2$	$k_{min} = 1.4$ $\eta = 0.95$	$\nu = 5.0 \times 10^{-3}$
	Median	3.288	3.253	**3.148**
	Mean	3.953	3.867	**3.73**

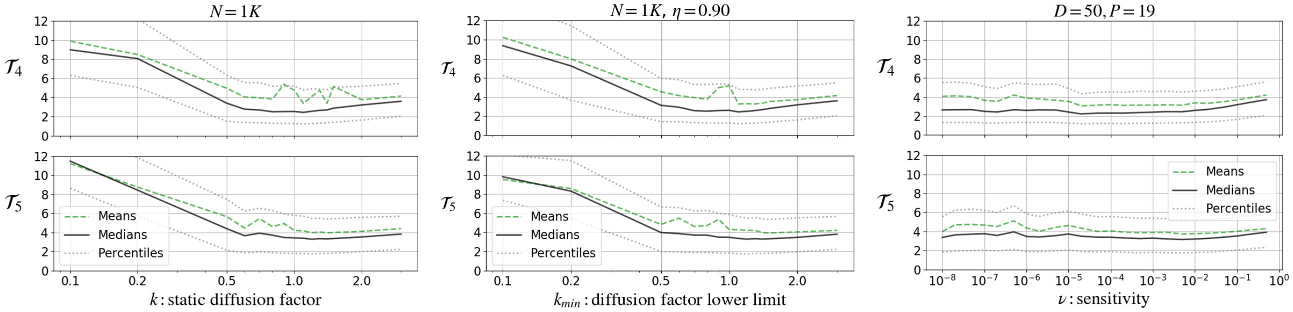
G. DISCUSSION

We compare our best error statistics that use real RSSI measurements with the state of art results that use mobile transmitters and stationary sensors in indoor experimental areas with similar sizes:

The work of Cantón Paterna *et al.* [19], is installed in an area of 270 m^2. They achieve an error rate of 4.6 m during 90% of the time using 4 computers and one beacon. They measure the performance of their work using interpolated points with respect to time stamps on predefined paths.

We do another comparison with a livestock tracking system [5]. The tests are conducted in a zone that houses 31 dairy Holstein cows with a size of 390 m^2. BLE transmitters are attached to the collars of the cows and RSSI data are captured using battery powered receivers. Moreover, they verify their results using a closed-circuit television (CCTV) system with three cameras. They measure an error median and mean of 3.3 m and 4.3 m respectively.

In terms of positioning performance, our error rate in the worst case scenario (3.148 m) is lower than the very similar setups. We observe that some of the experiments



(a) Static diffusion strategy (b) Decaying diffusion strategy (c) Adaptive diffusion strategy

FIGURE 17. Real RSSI data: Error (m) means and medians for diffusion factor strategies. The black, green and gray curves represent the median, mean and percentiles (25% and 75%) respectively.

yield higher error rates naturally, but we infer that these samples are not numerous, considering the low variances of the adaptive strategies (see FIGURE 17c). Moreover, we use a set of RSSI data annotated with its ground truth position using an AR based truth detection system, which makes the evaluation of our work more accurate compared to its counterparts.

The only downside of our work is the prior scene analysis stage based on fingerprinting. The performance depends on the number of reference points and durations, which makes this stage tedious. However, we also bring a solution to mitigate this workload with the Wasserstein histogram interpolation using lower number of reference points. Moreover, the AR based ground truth collection facility helps to automatically annotate the reference points without measuring them manually. The CCTV system used in the cow barn [5] does a similar online ground truth positioning of the cows, which is an indispensable requirement for measuring the performance correctly.

All of the works use low cost and affordable transmitters and sensors. Cantón Paterna *et al.* [19] use multiple low cost computers to expand the coverage, because higher number of sensors relates to better positioning performance, but usage of stationary computers for sensing brings extra cost. Trogh *et al.* [5] propose low cost and battery powered receivers. Likewise, we bring forth a scalable solution to expand the sensing coverage using low cost USB Bluetooth adapters and extension cables. The cables can be extended more as long as they can supply enough power, and a lower number of computers can host a swarm of Bluetooth adapters.

In summary, when compared to the works with similar setups, the results show that the proposed work achieves better positioning performance with accurate evaluation. Moreover, we do not only compare our localization algorithm with its accuracy. We also show that it is scalable for space, hardware and complexity situations, as new sensors and low cost computers can be added easily and the complexity may be tuned with different particle population sizes.

V. CONCLUSION

Bluetooth low-energy transmitters are energy efficient RF packet emitters. Bluetooth sensors sharing the same environment with the transmitters extract RSSI parameters from the captured packets, which can be used as position indicators, but very bad indicators because of both the multipath problem of the RF signal nature and the instability of the signal indicators through time. This work deals with the transmitter tracking problem that depends only on the RSSI data emitted by a mobile BLE beacon. Having no information on how the object moves, we model the motion of the tracked object as a diffusion in the environment and approach the problem with an adaptive sequential Monte Carlo filter.

The novelty of this work comes from an additional latent layer of the state space model that is used for online parameter sampling. Hence, Monte Carlo sampling is performed in two subsequent layers: firstly, diffusion particles are sampled and used as error covariance parameters, and in the second actual filtering layer position particles are sampled. We show that in a classical SMC filter, we should choose an error covariance considering a stable motion of the tracked object and the efficiency of the emission density model. This is highly inappropriate since the velocity of the object may vary and the emission density model can be very inaccurate.

The diffusion particles are sampled with their own error covariance, a sensitivity value. We show that this sensitivity value can be chosen from a large space compared to the point estimate of the static diffusion factor. With an almost arbitrary sensitivity value, the diffusion factors get adapted to the problem and more accurate position particles can be achieved. We also show that this method is efficient for both performance and accuracy with respect to the classical SMC filter. The contribution has potential to be applied to the localization environments where propagation conditions change significantly over time. Localization algorithms with fixed parameters fail in such environments, however the proposed solution is based on fingerprinting, which may bring a bottleneck due to the emprical parameters. We believe that this bottleneck can also be overcome with the application of the Wasserstein histogram interpolation.

In the proposed IPS, captured data by the distributed sensors are transferred to a single computer where the SMC filter algorithm is run. As a future work, the system can be made decentralized by running the SMC filter on each sensor unit. This work will require to determine the particle sizes the sensor units can process in real time, and for better approximation the posterior information should be efficiently shared between the processing nodes in terms of particles.

ACKNOWLEDGMENT

The authors would like to thank to Dr. A. Teoman Naskali for his invaluable collaboration in building the navigable platform and collecting data and PoiLabs for supplying POI Beacons for this work.

REFERENCES

[1] S.-C. Kim, Y.-S. Jeong, and S.-O. Park, "RFID-based indoor location tracking to ensure the safety of the elderly in smart home environments," *Pers. Ubiquitous Comput.*, vol. 17, no. 8, pp. 1699–1707, Dec. 2013.

[2] M. J. M. Chowdhury, M. S. Ferdous, K. Biswas, N. Chowdhury, and V. Muthukkumarasamy, "COVID-19 contact tracing: Challenges and future directions," *IEEE Access*, vol. 8, pp. 225703–225729, 2020.

[3] R. Brena, J. García-Vázquez, C. G. Tejada, D. Muñoz, C. Vargas-Rosales, J. Fangmeyer, Jr., and A. Palma, "Evolution of indoor positioning technologies: A survey," *J. Sensors*, vol. 2017, Mar. 2017, Art. no. 2630413.

[4] I. Bisio, A. Sciarrone, and S. Zappatore, "A new asset tracking architecture integrating RFID, Bluetooth low energy tags and ad hoc smartphone applications," *Pervasive Mobile Comput.*, vol. 31, pp. 79–93, Sep. 2016.

[5] J. Trogh, D. Plets, L. Martens, and W. Joseph, "Bluetooth low energy based location tracking for livestock monitoring," in *Proc. 8th Eur. Conf. Precis. Livestock Farming*, 2017, pp. 469–475.

[6] A. Xiao, R. Chen, D. Li, Y. Chen, and D. Wu, "An indoor positioning system based on static objects in large indoor scenes by using smartphone cameras," *Sensors*, vol. 18, no. 7, p. 2229, Jul. 2018.

[7] J. Biswas and M. Veloso, "Depth camera based indoor mobile robot localization and navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 1697–1702.

[8] G. Simon, G. Zachár, and G. Vakulya, "Lookup: Robust and accurate indoor localization using visible light communication," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 9, pp. 2337–2348, Sep. 2017.

[9] I. A. Koc, T. Serif, S. Gören, and G. Ghinea, "Indoor mapping and positioning using augmented reality," in *Proc. 7th Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2019, pp. 335–342.

[10] C. Medina, J. Segura, and A. Torre, "Ultrasound indoor positioning system based on a low-power wireless sensor network providing sub-centimeter accuracy," *Sensors*, vol. 13, pp. 3501–3526, Mar. 2013.

[11] G. Kumar, A. Patil, R. Patil, S. Park, and Y. Chai, "A LiDAR and IMU integrated indoor navigation system for UAVs and its application in real-time pipeline classification," *Sensors*, vol. 17, no. 6, p. 1268, Jun. 2017.

[12] W. C. S. S. Simões, W. A. E. Silva, M. M. D. Lucena, N. Jazdi, and V. F. D. Lucena, "A hybrid indoor positioning system using a linear weighted policy learner and iterative PDR," *IEEE Access*, vol. 8, pp. 43630–43656, 2020.

[13] S. Gezici, "A survey on wireless position estimation," *Wireless Pers. Commun.*, vol. 44, no. 3, pp. 263–282, Feb. 2008.

[14] A. Diallo, Z. Lu, and X. Zhao, "Wireless indoor localization using passive RFID tags," *Procedia Comput. Sci.*, vol. 155, pp. 210–217, Jan. 2019.

[15] H. Xu, Y. Ding, P. Li, R. Wang, and Y. Li, "An RFID indoor positioning algorithm based on Bayesian probability and K-Nearest neighbor," *Sensors*, vol. 17, no. 8, p. 1806, Aug. 2017.

[16] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. Al-Ammar, and H. Al-Khalifa, "Ultra wideband indoor positioning technologies: Analysis and recent advances," *Sensors*, vol. 16, no. 5, p. 707, May 2016.

[17] P. Kriz, F. Maly, and T. Tomáš, "Improving indoor localization using Bluetooth low energy beacons," *Mobile Inf. Syst.*, vol. 2016, pp. 1–11, Apr. 2016.

[18] G. Han, H. Xu, T. Q. Duong, J. Jiang, and T. Hara, "Localization algorithms of wireless sensor networks: A survey," *Telecommun. Syst.*, vol. 52, no. 4, pp. 2419–2436, Apr. 2013.

[19] V. C. Paterna, A. C. Augé, J. P. Aspas, and M. P. Bullones, "A Bluetooth low energy indoor positioning system with channel diversity, weighted trilateration and Kalman filtering," *Sensors*, vol. 17, no. 12, p. 2927, Dec. 2017.

[20] F. Daniş and A. Cemgil, "Model-based localization and tracking using Bluetooth low-energy beacons," *Sensors*, vol. 17, no. 11, p. 2484, Oct. 2017.

[21] I. Bisio, F. Lavagetto, M. Marchese, and A. Sciarrone, "Smart probabilistic fingerprinting for WiFi-based indoor positioning with mobile devices," *Pervasive Mobile Comput.*, vol. 31, pp. 107–123, Sep. 2016.

[22] A. Doucet, N. de Freitas, and N. Gordon, *An Introduction to Sequential Monte Carlo Methods*. New York, NY, USA: Springer, 2001, pp. 3–14.

[23] O. Cappé, E. Moulines, and T. Ryden, *Inference in Hidden Markov Models* (Springer Series in Statistics). Berlin, Germany: Springer-Verlag, 2005.

[24] H. Nurminen, A. Ristimäki, S. Ali-Löytty, and R. Piché, "Particle filter and smoother for indoor localization," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat.*, Oct. 2013, pp. 1–10.

[25] R. M. Shubair and H. Elayan, "Enhanced WSN localization of moving nodes using a robust hybrid TDOA-PF approach," in *Proc. 11th Int. Conf. Innov. Inf. Technol. (IIT)*, Nov. 2015, pp. 122–127.

[26] P. Filípek and A. Kovarova, "Indoor localization based on beacons and calculated by particle filter," in *Proc. 17th Int. Conf. Comput. Syst. Technol. (CompSysTech)*. New York, NY, USA: Association for Computing Machinery, Jun. 2016, pp. 269–276.

[27] H. Lang, T. Li, G. Villarrubia, S. Sun, and J. Bajo, "An adaptive particle filter for indoor robot localization," in *Ambient Intelligence—Software and Applications*, A. Mohamed, P. Novais, A. Pereira, G. V. González, and A. Fernández-Caballero, Eds. Cham, Switzerland: Springer, 2015, pp. 45–55.

[28] D. Fox, "KLD-sampling: Adaptive particle filters," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst. Natural Synth. (NIPS)*. Cambridge, MA, USA: MIT Press, 2001, pp. 713–720.

[29] A. D. Bagdanov, A. Del Bimbo, F. Dini, and W. Nunziati, "Adaptive uncertainty estimation for particle filter-based trackers," in *Proc. 14th Int. Conf. Image Anal. Process. (ICIAP)*, Sep. 2007, pp. 331–336.

[30] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. Amer. Stat. Assoc.*, vol. 94, no. 446, pp. 590–599, Jun. 1999.

[31] S. Yıldırım, S. S. Singh, T. Dean, and A. Jasra, "Parameter estimation in hidden Markov models with intractable likelihoods using sequential Monte Carlo," *J. Comput. Graph. Statist.*, vol. 24, no. 3, pp. 846–865, Jul. 2015.

[32] N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski, and N. Chopin, "On particle methods for parameter estimation in state-space models," *Stat. Sci.*, vol. 30, no. 3, pp. 328–351, Aug. 2015.

[33] J. Yun, F. Yang, and Y. Chen, "Augmented particle filters," *J. Amer. Stat. Assoc.*, vol. 112, no. 517, pp. 300–313, Jan. 2017.

[34] N. Chopin, "A sequential particle filter method for static models," *Biometrika*, vol. 89, no. 3, pp. 539–552, Aug. 2002.

[35] A. Papavasiliou, "Parameter estimation and asymptotic stability in stochastic filtering," *Stochastic Processes their Appl.*, vol. 116, no. 7, pp. 1048–1065, Jul. 2006.

[36] N. Chopin, P. E. Jacob, and O. Papaspiliopoulos, "SMC2: An efficient algorithm for sequential analysis of state space models," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 75, no. 3, pp. 397–426, Jun. 2013.

[37] D. Crisan and J. Míguez, "Nested particle filters for online parameter estimation in discrete-time state-space Markov models," *Bernoulli*, vol. 24, no. 4A, pp. 3039–3086, Nov. 2018.

[38] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *J. Comput. Graph. Statist.*, vol. 5, no. 1, pp. 1–25, Mar. 1996.

[39] T. Li, M. Bolic, and P. M. Djurić, "Resampling methods for particle filtering: Classification, implementation, and strategies," *IEEE Signal Process. Mag.*, vol. 32, no. 3, pp. 70–86, May 2015.

[40] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, 2009, vol. 3, no. 3.2, p. 5.

F. SERHAN DANİŞ received the B.Sc. degree from the Department of Computer Engineering, Galatasaray University, in 2006, and the M.Sc. degree from the Department of Systems and Control Engineering, Boğaziçi University, in 2009, where he is currently pursuing the Ph.D. degree with the Department of Computer Engineering. He has been working as a Research Assistant at Galatasaray University, since 2007. His research areas include Bluetooth low-energy localization and tracking, and sequential Monte Carlo methods and Bayesian statistics.

A. TAYLAN CEMGİL received the Ph.D. degree from Radboud University Nijmegen. He is currently a Professor with the Department of Computer Engineering, Boğaziçi University, Istanbul, Turkey. Before that, he held a postdoctoral position with the University of Amsterdam and Cambridge University. He worked in collaboration with the industry on several applications, such as audio and music processing, anomaly detection, tracking, sensor fusion, recommendation systems, and customer analytics. His research interests include basic machine learning methodologies, in particular robust machine learning, Bayesian modeling/inference, Monte Carlo methods, and matrix/tensor decompositions.

CEM ERSOY (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical and electronics engineering from Boğaziçi University, Istanbul, in 1984 and 1986, respectively, and the Ph.D. degree in electrical engineering from Polytechnic University of New York, Brooklyn, NY, USA, in 1992. He worked as an Research and Development Engineer with Northern Telecom subsidiary NETAS A.S. Since then, he has been a Professor with the Computer Engineering Department and the leader of the Wireless Sensor Networks Research Group, Boğaziçi University. His research interests include wireless networks, activity recognition and ambient sensing, pervasive health, multi-tier edge computing, SDN/NFV, the IoT, and green networking.