SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 3, 410-421, 2025
DOI: 10.35377/saucis...1644762

RESEARCH ARTICLE

Radio-Frequency Map Optimization for Indoor Positioning and Tracking

F. Serhan Daniş[1,2,*]

[1]Galatasaray University, Computer Engineering Department, 34349 Istanbul, Türkiye, ror.org/00btgsb62
[2]Gustave Eiffel University, AME-GEOLOC, 44380, Bouguenais, France, ror.org/03x42jk29

Corresponding author:
F. Serhan Daniş,
Galatasaray University, Gustave Eiffel University
sdanis@gsu.edu.tr

Article History:
Received:21.02.2025
Revised:06.07.2025
Accepted:22.07.2025
Published Online: 24.09.2025

ABSTRACT
We introduce a parameter optimization strategy to enhance the accuracy of an indoor positioning system. The indoor positioning system of interest is composed of subsequent stages of processing of reference reduced signal strength indicator (RSSI) streams, Gaussian processes-based estimation of probabilistic radio-frequency maps, and an adaptive particle filter that is used to infer the trajectory of the tracked object. Each stage has its own model parameters, which can be evaluated by the accuracy of the final trajectory estimations given their ground-truth counterparts. We make use of an open dataset that includes RSSI data on reference points, RSSI data related to trajectories and their corresponding ground-truth positions. By being able to evaluate the estimations, we develop a Monte Carlo particle swarm optimization strategy to search for the best parameter configuration that minimizes the trajectory error. The time performance of the optimization strategy is also improved by artificially discretizing the parameters space, so that the stages can use the previously processed streams or radio maps. We show that the strategy can both improve accuracy and decrease the search time with respect to a grid-based search strategy.
Keywords: Fingerprinting, Indoor Positioning, Monte Carlo Swarm Optimization, Radio Frequency Map Estimation, Reduced Signal Strength Indicator

1. Introduction

Indoor Positioning Systems (IPS) have become an interestingly popular topic in recent decades, as people began to pass their time in enclosed places like shopping malls, airports, or hospitals, where Global Navigation Satellite System (GNSS) signals are unreliable, and positioning is consequently inaccurate. The domain has produced various techniques based on already existing technologies like Wi-Fi [1], Bluetooth Low-energy (BLE) [2], ultra-wideband (UWB) [3], visible-light or acoustics [4]. While there is still no standardization of the concept, it has a broad range of already installed and future applications across different sectors. Some examples may be assisting people in navigating large buildings, and especially visually impaired individuals [5]; tracking equipment, inventory or assets in warehouses, hospitals, or factories; enhancing shopping experience by delivering location-based promotions to customers; and most importantly, helping emergency services locate living beings in complex buildings in emergency situations. Technology themselves introduce challenges due to phenomena like signal interference and multipath [6]. Combined with these natural challenges, varying layouts and materials of the indoor environments and varying sensitivities of the devices make signal propagation unpredictable. These challenges can be amended by fingerprinting at the expense of initial setup costs, data collection and frequent updates of the database.

In the domain of indoor positioning, fingerprinting is a time-consuming process that is used to capture a gist of the signal parameter distributions specific to a closed space [7]. The distributions are obtained by recording the data and measuring their statistics on some reference positions (RP) in the area of interest. This analysis, prior to the actual tracking algorithm, is shown to perform better than its counterparts like multilateration [8]. The collected signal parameter distributions are compiled into databases in terms of radio-frequency maps (RM), which are in turn used to measure the likelihood of a signal parameter at a specific position. In the simplest manner, the positions of live measurements can be inferred by comparing them to signal distributions on RPs [9]. The accuracy of this direct comparison is bound to the number of selected RPs. However, increasing the number of RPs leads to collecting a high amount of data. Therefore, accurate RM estimation approaches that use smaller sets of RPs are favored, alongside the automated data collection techniques [10].

Researchers prefer to interpolate the signal parameter distributions over the unknown positions of the area, assuming that a distribution will be similar to the distributions in the neighborhood of it. By properly interpolating the distributions, we produce a signal parameter distribution on any position, which will be called an RM estimate of the area. In the work of Nabati and Ghorashi [11], the distribution of available signal strength samples is learned using a deep neural network (DNN).

Cite as: F. S. Daniş. (2025). Radio-frequency map optimization for indoor positioning and tracking. *Sakarya University Journal of Computer and Information Sciences*. vol. 8, no. 3, pp. 410-421, 2025 doi: 10.35377/saucis... 1644762

This makes the approach bound solely to the data. Such data-driven approaches like DNN introduce challenges too due to high locality [12]. Smoother model-based approaches are proposed to mitigate these challenges. In Zhao et al. [13], the authors consider the received signal strength indicator (RSSI) fingerprints as incomplete and data distributions on unknown RPs are estimated based on Gaussian processes (GP) using a limited number of surrounding RPs. They first estimate the model parameters of the GPs through maximum likelihood estimation (MLE) by assuming that all positions in their training set are precisely known. Along with their interpolation purposes, GPs present a measurement of the uncertainty for the positions [14].

In this work, we augment a fingerprint-based positioning system by integrating it with the fingerprinting processing. The final positioning system is composed of three major stages: (i) a live RSSI signal preprocessing, (ii) an RM estimation stage and (iii) a particle filter-based position/trajectory estimator. In the signal processing stage, the RSSI data are exposed to a maximal filter, which was shown to increase the accuracy significantly [15]. In the RM estimation stage, the filtered position-bound RSSI data are processed into position-bound distributions, which are then interpolated with GPs to build an RM of a dense grid of positions. The RM estimation step is succeeded by the actual positioning algorithm based on an adaptive particle filter that employs the RM estimates as emission densities [16]. Each stage has its specific parameters that need to be optimized. The accuracy of the final trajectory estimator does not only depend on the parameters of the positioning algorithm, but also on the parameters of the preceding stages. We consider these system parameters as latent variables which are hard to set heuristically.

Considering the entire system, we can evaluate the accuracy once it is supplied with a configuration of the parameter values. We benefit from a dataset that is compiled with trajectory-bound RSSI values and their corresponding ground-truth positions [17]. The access to precise ground-truth information enables us to automate the evaluation of the subprocesses by iteratively varying the parameter values. In the previous studies, the parameter search was limited to the separate stages and was merely a naïve grid search, which would spend time evaluating parameter values far from suboptimal values [9]. We introduce a state vector whose values are used to set the values of the stages of the IPS. As we can evaluate the value of this state vector by running the entire system, we convert the problem into an optimization problem, in which we search for the optimum value of the state vector. If the evaluation function were differentiable, then gradient-based iterative methods could be employed [18]. As the proposed system is far from being differentiable, we make use of the gradient-free methods, in which we assume that we have access to pointwise evaluation of the search space. However, measuring the error of each configuration corresponds to repeating the computationally heavy stages for each trajectory.

The two options of gradient-free optimization are the random-walk search or numerical approximation of the gradient search [19], for the first of which convergence is unlikely with continuous and high-resolution parameter spaces. The second option, the numerical approximation of the gradient search, requires a systematic neighborhood exploration around a candidate point. However, because of the computational heaviness of the stages and the dimension of the state space, the neighborhood exploration is not tractable in high-dimensional spaces. To mitigate the heavy load of the parameter search, we benefit from a random walk with multiple candidates in a smart, discretized setup. We introduce a Monte Carlo particle swarm optimization (MCPSO) in an artificially discretized parameter space. The discretization is performed as a dense but finite grid of parameters, by which we also keep the intermediate outputs of the best parameters so far, to avoid repetitions of the computationally intensive intermediate steps to a degree. The swarm optimization strategy iteratively updates them by converging on the best parameter vector. The strategy is greedy and is not guaranteed to converge to the optimum configuration, but we achieve faster implementations and more accurate results with respect to a grid search.

The two contributions of this work are the proposed MCPSO strategy and the GP-based RM estimation for indoor positioning purposes. Section 2 explains these two strategies and their parameter space. In Section 3, the experiments and their results are shown. We finally conclude in Section 4 with potential future directions.

2. Methodology

We propose an iterative RM optimization system that aims to find the best parameter configuration of a fingerprinting-based IPS. The general overview of the optimization system is given in Figure 1. The main part of the optimization system is the IPS, which is composed of three subsequent stages: the signal processing, the RM estimator, and the trajectory estimator. Each of these stages has its own model parameters that are brought together to form a single state vector, a value of which is evaluated by running the positioning system to generate a trajectory estimate and comparing it with the corresponding ground-truth trajectory. The error statistics between the two are fed as optimization feedback to select a new value for the state vector.

In the IPS, the RP-bound RSSI streams are first exposed to a “Signal Processing” stage that employs the maximal filter, which processes the RSSI signals and produces maximalized RP-bound RSSI histograms for the RPs. In the filter, the last RSSI value of a time window of size, , is replaced by the highest RSSI value in that window. The intuition behind this filter is that the highest RSSI value is expected to be the measurement of the line-of-sight communication or the communication with the largest reflection angle. The lower values are regarded as the reflections with lower angles, so that these BLE packets have lower signal strengths with respect to the one that arrives with the largest angle. The histograms of the maximalized RP-bound RSSI streams have exceptionally low variances. This leads us to approximate the RSSI histogram with a single-moded probability distribution and keep its mean to define the corresponding RP. Thus, the output of the signal processing step is a list of single RSSI values per RP on the map. The variances of the approximated distributions are not taken into consideration.

In the "RM Estimator" stage, the obtained RP-bound RSSI means on the RPs are used to produce RMs through interpolation, for which we employ the Gaussian Process Regression (GPR). The details of the GPR for generating RM estimates are given in Section 2.1. The output of this stage is a grid of RSSI values, or, namely, an estimate for the radio-frequency map. We achieve a spectrum of RM estimates by varying the GPR parameters.

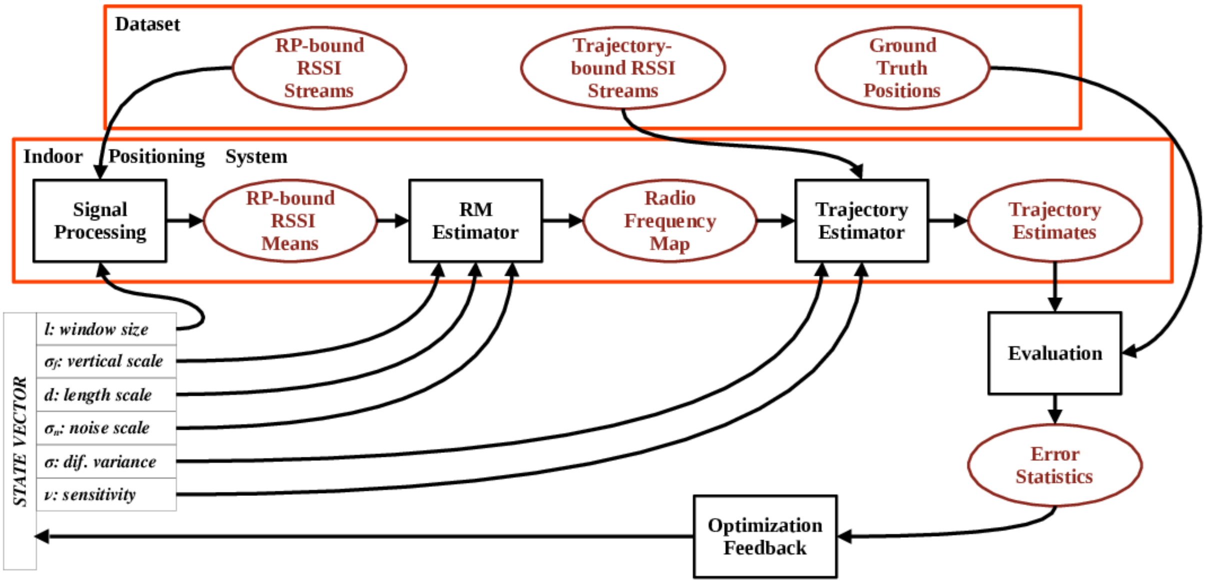


Figure 1. Optimization system overview

The "Trajectory Estimator" stage uses two data sources: (i) an RM estimate from the previous stage, and (ii) the RSSI data recorded by navigating the BLE emitter in the environment. We must note that the RSSI data of the navigation is also exposed to the maximal filter with the same window size as the one used to generate the RM. For the trajectory estimator, we employ a previously proposed adaptive particle filter. We obtain position estimates belonging to a trajectory for the given RSSI data. The estimation system is driven by two parameters. The sensitivity was introduced for controlling the transition density of the particle filter by sampling both the particles and their diffusion-based predictions at the same time. Here, we introduce another parameter, the diffusion variance, to define a fixed variance of the Gaussian distributions on each position of the map. The RM estimate with the diffusion variance is used as the emission density, and the RSSI data are fed to the filter as observations. The positioning system yields these timestamped position estimates belonging to a trajectory as output.

With their ground-truth correspondences in hand, we can compute the error between a trajectory estimate and its true value. In the "Evaluation" stage, we yield the statistics of these errors for each available trajectory-bound RSSI data stream. From these statistics, we pick the mean as the "Optimization Feedback" to regenerate a new state vector value. The details of the optimization process are given in Section 2.2.

2.1 Gaussian Process Regression for RM Estimation

Gaussian processes are considered to interpolate intermediate values when there are few and noisy data points as references from a system, which corresponds to our case of the fingerprinting-based positioning [20]. To make an RM estimation, we state the problem as an estimation of RSSI distributions on arbitrary map positions. Given the RP coordinates and their associated RSSI values, we are interested in estimating an RSSI distribution at any point. With the introduction of the signal processing stage, we can assume that these distributions are Gaussian and formulate the problem as a Gaussian process [21].

We regard the associated RSSI values on the RPs as observations. The relation between the observations is achieved with the covariance function (or the kernel function). We use the two popular choices for the kernel function: (i) the absolute exponential, or the -distance and (ii) the squared exponential, or the -distance. The mathematical formulations of these kernels [21] are given in (1) and (2), respectively:

$$k_l\left(\mathbf{x},\mathbf{x}'\right) = \sigma_f^2 \exp\left[\frac{-\left|\mathbf{x}-\mathbf{x}'\right|}{d}\right] + \sigma_n^2 \delta\left(\mathbf{x},\mathbf{x}'\right) \quad (1)$$

$$k_2\left(\mathbf{x}, \mathbf{x}'\right) = \sigma_f^2 \exp\left[\frac{-\| \mathbf{x} - \mathbf{x}' \|_2^2}{d}\right] + \sigma_n^2 \delta\left(\mathbf{x}, \mathbf{x}'\right) \tag{2}$$

According to these models, we expect the kernel to yield higher values for functions that cover a broad range on the input domain. If two points (x and x') are close, then $k(x, x')$ is high. Otherwise, the function value tends to zero, which means that distant observations will not affect the points to be interpolated. We calculate the covariance matrix, K, along with the effect of the points of interpolation, K_* and K_{**}.

The observations are expected to be noisy; that is, the RSSI value that is observed from a certain RP may not define that RP accurately. To take into account different noise variations, we employ a version of GPR that is driven by three hyper parameters: the vertical scale or the signal variance, σ_f, which determines the magnitude of the fluctuations in the RSSI domain, the length scale or the horizontal scale, d, which indicates how quickly the correlation of two points of the map is reduced as their distance increases, and the noise variance, σ_n, which quantifies variability in the target variable or the level of uncertainty in the observed data points [22]. The estimated value, $\tilde{y}$ at a point$x$is distributed along a multivariate Gaussian distribution given the previous measurements:

$$\tilde{y} \mid y \sim \ (K_* K^{-1} y, K_{**} K^{-1} K_*^T) \tag{3}$$

The mean of the equation (2) is a good estimate for the specific position. The goodness can also be measured with the variance of the same distribution. By varying the map coordinates, x, we generate different estimates of RSSI values on a dense grid, which serves as the RM of the area. Samples of these RMs are shown in Figure 2.

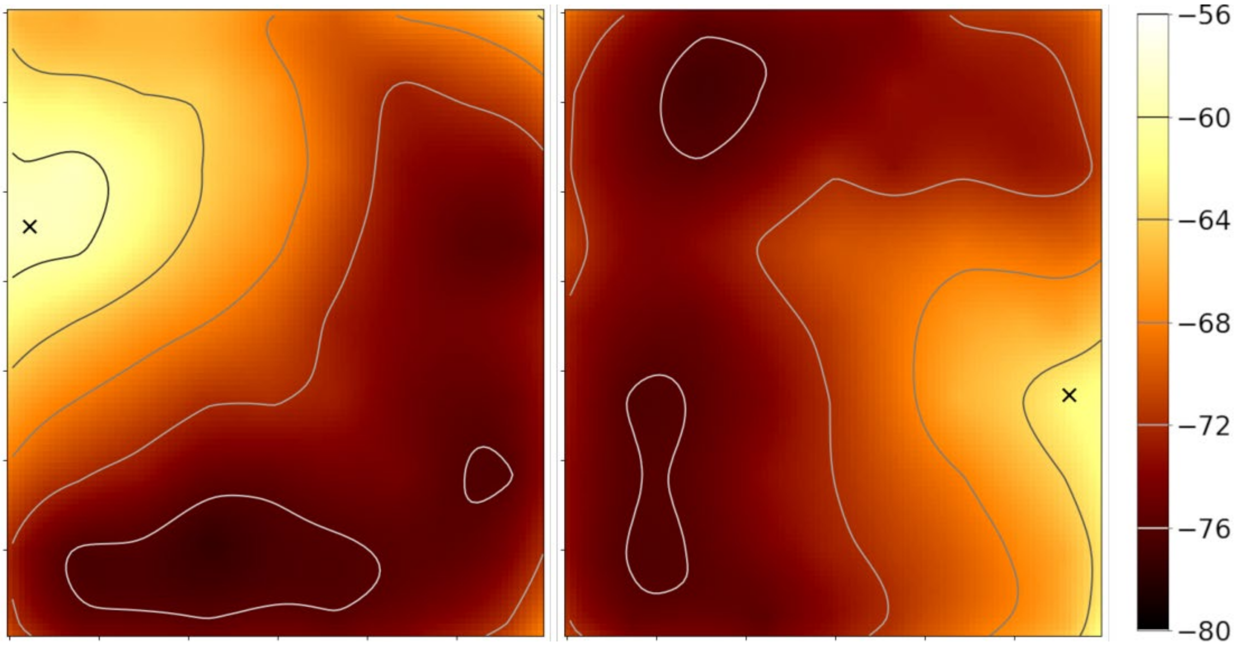


Figure 2. Radio-frequency map estimate examples. The points × are the sensor positions.

2.2. Monte Carlo Particle Swarm Optimization

We design a sampling-based iterative optimization strategy to search for the optimal values of a state vector, which embodies the parameters of the successive stages of an IPS, as given in Figure 1. The evaluation of a state vector value is achieved by passing through the stages and comparing the position estimates with the ground-truth positions. We call the iterative search strategy Monte Carlo Particle Swarm Optimization (MCPSO). Given an RSSI data stream from the dataset we want to minimize the positioning error produced with a value of the state vector, which will be denoted by $s_t^i = (\ell, \sigma_f, d, \sigma_n, \nu)_t^i$ where t is the iteration number, and $i$is the index in the population. At each iteration, a population of state vectors, S_t, is generated and evaluated. We will denote the population size with P, the decay factor with η, the size of the best state vector samples with E, and the covariance of the sampling distribution with κ_t.

Initialization: The initial population of values is sampled uniformly between the given extremities of the search space: $s_t^i \longleftarrow U(S)$. Along with the population initialization, we also set the initial covariance, $\kappa_t = C$.

Evaluation: At each iteration t, the successive stages of the IPS system are repeated for each individual state vector in the population. We denote the function of the IPS system as $\mathcal{F}(s_t^i)$, after which we obtain a trajectory estimate. Each trajectory estimate is compared with the ground-truth trajectory, Ω. The comparison is performed by measuring the Euclidean distance of the individual positions of the trajectory to their true correspondences. We obtain a set of positioning errors after this evaluation, and we take the mean of these errors as the performance measure of the current individual state vector. The state vector individuals are reordered with respect to their error means, from which a new set of elite vectors, $\tilde{S}_t$, of size E, is formed. The first state vector value in this set is also stored as the best value so far.

Repopulation: The state vector individuals in the elite set, $\tilde{S}_t = \{\tilde{s}_t^i\}$, are used to repopulate the next population S_{t+1}. The generation procedure is based on taking samples from a Gaussian distribution whose mean is set to a randomly chosen elite state vector individual:

$$\tilde{s}_{t+1}^i = N(\tilde{S}_t, \kappa_t)$$

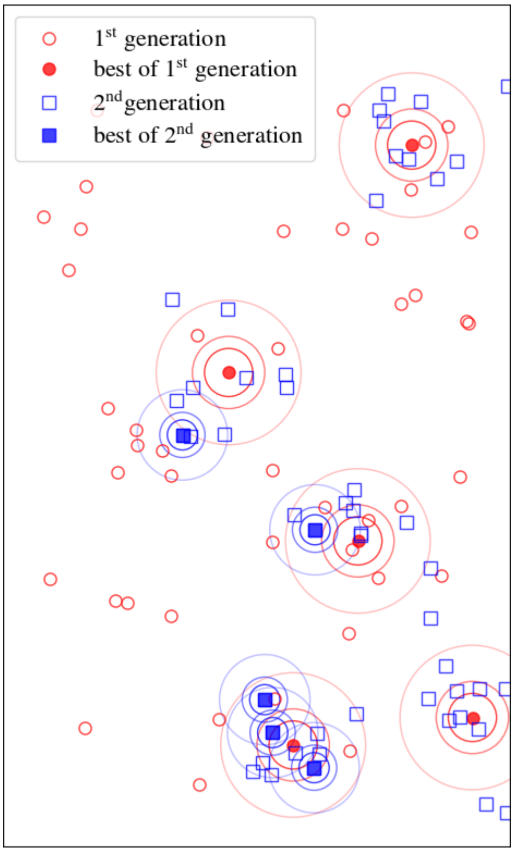
$$\tilde{S}_{t+1} = \{s_{t+1}^i\} \cup \tilde{S}_t$$

where the covariance matrix of the sampling distribution, κ_t, is spherical. We also adopt an elitist approach and directly transfer these elite state vectors to the next iteration.

Inputs
Ground-truth positions: Ω
Population size: P
Elites size: E
Decay factor: η
Initial covariance: $\mathbf{C}$
Output:
Optimal state vector: $\mathbf{s}^*$
Minimum error statistics: ϵ

1: **Initialize**:
  $t \leftarrow 0,\ \boldsymbol{\kappa}_0 \leftarrow \mathbf{C}$
  for $i \in 1 \dots E$ **do**
    $\mathbf{s}_t^i \leftarrow \mathcal{U}(\mathcal{S})$
2: **Evaluate and select**
  $\tilde{\mathbf{S}}_t \leftarrow \arg\min^{1 \dots E}(|\mathcal{F}(\mathbf{s}_t^i) - \Omega|)$
  $\mathbf{s}^* \leftarrow \tilde{\mathbf{S}}_t^0$
3: **Repopulate with elites**
  for $i \in 0 \dots P$ **do**
    $\mathbf{s}_{t+1}^i \leftarrow \mathcal{N}(\tilde{\mathbf{S}}_t, \boldsymbol{\kappa}_t)$
  $\mathbf{S}_{t+1} \leftarrow \{\mathbf{s}_{t+1}^i\} \cup \tilde{\mathbf{S}}_t$
4: **Recurse**
  if not converged:
    $t = t + 1,\ \boldsymbol{\kappa}_t = \eta . \boldsymbol{\kappa}_{t-1}$
    Go back to 2.
  else
    Return $\mathbf{s}^*$, ϵ
  end

(a) MCSPO Algorithm



(b) An illustration of the MCSPO in 2D space.

Figure 3. Monte Carlo Particle Swarm Optimization (MCPSO) for Radio-Frequency Map Estimation

Recursion: When the convergence conditions are not met, the covariance matrix is reduced by a decay factor, η, so that the focus of the search space is gradually narrowed onto the current elite state vector values. The same procedure is repeated until convergence, that is, when no change is achieved for a number of successive iterations. Amongst all the state vector values, the value that generates the lowest mean error, s^*, is returned as the best output, along with the mean error itself, ϵ.

The pseudocode of the algorithm is given in Figure 3a. As a summary, a generation of particles is initialized uniformly randomly in a predefined space, which is six-dimensional in our case. An illustration that simulates in 2D is also provided in Figure 3b, in which the red particles (○) stand for the first generation. After an evaluation by the system, the best of the particles of the first generation (●) are chosen to generate a new generation (□) with respect to a Gaussian distribution around them. A new set of best particles (■) is selected after being evaluated by the system. The algorithm iterates in a recursive fashion until convergence. The underlying intuition of the algorithm is the belief that the space state is composed of many local minima. We let the search keep the diversity with the initial populations. By decaying the covariance factor, we gradually focus the population onto the best set of these local minima.

3. Experiments and Results

The proposed optimization system is possible only with the availability of a specific dataset. As shown in Figure 1, the portion of the dataset used in this work consists of three distinct types: (i) the RP-bound RSSI streams are collected by putting the beacon stationary on known reference positions (RP) on the map. (ii) Timestamped trajectory-bound RSSI streams are collected by navigating the beacon on predefined trajectories. (iii) High-resolution ground-truth positions are also collected alongside the RSSI streams of these trajectories. The dataset is compiled using 12 sensors that capture the packets emitted by a mobile BLE beacon in an office area of 364m². The details of the dataset and the data collection procedure are documented in Daniş *et al.* [17].

The method of this work makes use of the RSSI data and their ground-truth correspondences. Even though these positions were originally used for evaluating the estimated trajectory positions of the indoor positioning algorithm in previous works, they also enable us to fully automate the optimization iterations in a supervised manner without any human intervention.

3.1. Experimental Design and Implementation Remarks

The experiments are performed on a server with two processors of type Intel(R) Xeon(R) CPU E5-2697A v4 @ 2.60GHz that run on 64 threaded cores in total. The disks used to write data throughout the optimization process are Micron 1100 Solid State Disks connected through a Serial ATA interface and mapped as a RAID5 array. Their writing and reading speeds are 500 MB/s and 530 MB/s, respectively. The MCPSO algorithm is implemented using Python 3.9. As stated in Section 2, each iteration consists of running the different stages of RSSI stream preprocessing, RM estimation and trajectory estimation. The important caveat of the optimization system is that the subsystems take a high amount of time. Whereas the trajectory estimation is designed to run in real-time by using previously generated RM maps, the optimization system must regenerate the RM maps from scratch in the search process. The processing durations also include reading, processing and writing times of the corresponding data.

In the optimization trials, we chose different configurations of the parameter values for the decay rate, $\eta \in \{0.95, 0.90, 0.85, 0,80, 0.75\}$; for the population size, $P \in \{50, 30, 15, 5, 1\}$; for the elite size, $E = 8$; for the initial covariance matrix, $C = 0.5 I_2$, and we set a convergence condition after 8 iterations in a row with no update in the elite vector values.

The signal processing subsystem takes full streams of timestamped RSSI data as input. Each RP possesses 12 RSSI stream files, as many as the number of sensors. These streams are processed into distribution parameters per reference point. In the RM estimation, the distribution parameters are read, processed, and written as distribution parameters on each grid position of the RM. The trajectory estimation subsystem reads the RM, along with the RSSI data stream related to a trajectory, to estimate the positions. The positioning algorithm of the trajectory estimation is run in 50 parallel processes to generate different timestamped position estimates, which are in turn evaluated against the ground-truth to generate error statistics for the given state vector. The mean durations of the individual stages are reported as 28.79, 12.15, 63.5 and 0.31 seconds for the signal processing, the RM estimation, the trajectory estimation and the evaluation, respectively. We must note that the duration of the signal processing step depends on the length of the reference files, which, in our case, corresponds to an average of 30 minutes of data capture duration or 40k data points per RP and 81 RPs in total. The duration of the RM estimation step is related to the size of each grid cell, which is taken as 0.2 m in this work. The trajectory estimation employs an adaptive particle filter, which is tuned with several parameters like the particle size (1000) and resampling method (systematic). These parameters are chosen to run the trajectory estimation process in real-time. Thus, the duration of the trajectory estimation process is determined by the length of the RSSI stream, which is itself 58 seconds. The leftover duration is the overhead that includes the initialization and the synchronization of the parallel runs. Finally, the evaluation step depends again on the length of the RSSI stream of the trajectory.

The search space of the state vector is defined by the following extremities of the parameters: $\ell \in [0.5, 0.75]$, $\sigma_f \in [0.1, 100]$, $d \in [0.1, 20]$, $\sigma_n \in [0.1, 20]$, $\nu \in [10^{-8}, 10^{-1}]$, and $\sigma \in [0.1, 20]$. These extremities of the search space are selected to cover the largest appropriate set in which the individual value is effective. As an example, the window size of the maximal filter, ℓ was shown to yield significantly higher errors for values lower than 0.5 due to the insufficient number of samples [15].

To speed up the search process, we design the fingerprint optimization procedure as an iterative search problem in a discretized state space, although the parameter space is originally continuous. We force the space to be divided into a high-resolution set of predefined floating numbers that encompasses a sufficiently large configuration space. Each dimension of the search space is redefined to have 1000 distinct points, which corresponds to a space of 10^{18} distinct states. By discretizing the search space, the window size in the signal processing subsystem can be reused if the corresponding reference histograms are previously generated. Accordingly, the radio-frequency map does not need to be regenerated if the corresponding Gaussian process parameters are selected for a second time. Instead, previously calculated errors are reused. This approach increases the time performance of the algorithm by repeating the subsystem parameters, which in turn totally avoids the execution of a subsystem. As the algorithm is iterated, the odds of selecting the exact state vector are also increased, due to the decreasing search covariance. If the same state vector is encountered, no subsystems are executed. The search is particularly quickened as error converges, or the sampling covariance decays. This enforcement of discretization enables the system to bypass some of the computationally heavy subsystems if the corresponding RPs or RMs are already computed in the previous iterations. Thus, the initial iterations take a considerable time, but as the optimization converges, more bypasses are possible.

As an important implementation remark, the grids of the parameters are kept as string data types to avoid the misinterpretation of the floating numbers. The approach removes the ambiguity while checking if the data of interest has been previously processed.

3.2. Preliminary Results

From the dataset, we choose four trajectory-bound RSSI streams: a zigzag ($\mathcal{E}_1$), a rectangular ($\mathcal{E}_2$) and two straight trajectories ($\mathcal{E}_3$ and $\mathcal{E}_4$). The preliminary experiments are run with $\eta = 0.80$. We first report the best positioning accuracies in Tables 1 and 2 for the two proposed kernel types of the Gaussian process regression model, L_1-norm and L_2-norm respectively.

Table 1. Best error statistics (m) and the corresponding parameter configuration for an L_1-norm kernel.

Exp.	ℓ	σ_f	d	σ_n	ν	σ	Median	Mean
$\mathcal{E}_1$	2.92	41.9	5.82	16.77	0.0002	3.59	0.795	0.989
$\mathcal{E}_2$	3.64	31.9	5.52	5.86	0.007	2.71	1.176	1.372
$\mathcal{E}_3$	0.98	22.5	19.80	16.91	0.0002	3.35	0.793	0.86
$\mathcal{E}_4$	1.82	35.5	18.51	9.12	0.002	3.23	0.95	1.719
All	2.27	43.8	13.59	15.70	0.0003	4.04	1.04	1.359

Table 2. Best error statistics (m) and the corresponding parameter configuration for an L_2-norm kernel.

Exp.	ℓ	σ_f	d	σ_n	ν	σ	Median	Mean
$\mathcal{E}_1$	1.4	84.5	4.05	13.27	0.0001	4.26	0.979	1.114
$\mathcal{E}_2$	3.19	79.0	4.02	7.67	0.004	3.27	1.068	1.289
$\mathcal{E}_3$	1.3	85.1	12.89	8.73	0.0002	3.29	0.704	0.827
$\mathcal{E}_4$	2.05	78.3	3.51	19.42	0.002	3.75	0.846	1.646
All	2.72	48.3	5.14	9.66	0.0005	4.5	1.107	1.42

Globally, there does not exist a significant distinction between the norms we use for the kernel of GPR in the context of RM estimation for indoor positioning. The error medians and means lie around 1 and 1.5 meters, respectively. In the specific experiments of $\mathcal{E}_1$ and $\mathcal{E}_3$, we achieve even lower error values, at the order of 0.7 and 0.9 meters. However, there is high variance among the best values of the parameters. Even though we report the best parameter configurations and the results of the individual experiments, the global errors are valuable as they pave the way to the generalization of the parameters. These global results, denoted in the tables as "All", are obtained by iterating the MCPSO on all the experimental trajectories. According to these results, we see that the median and the mean overall positioning errors can be reduced to 1.04 and 1.359 meters, respectively.

The errors do not vary between the two norm selections; however, the parameters vary significantly. A window size (ℓ) of the maximal filter can be chosen between 1.0 and 2.5 seconds. The specific parameters of the Gaussian processes vary with respect to the kernel type: for the L_1-norm kernel, the maximum allowed covariance, σ_f, resides around 35, and the scale of the noise term, σ_n, around 10. The same values of the L_2-norm kernels are respectively 80 and 15. We also observe that the length scale, d, varies too much, 13 and 5 meters. We interpret that for an L_1-norm kernel, the measurements of the RM are affected by the reference points in a larger area, whereas, for an L_2-norm kernel, this influence is narrower. Moreover, although the filtered RSSI histograms display low-variance histograms whose variances can be coarsely inferred to be at the scale of 2, we show that higher common diffusion variances, σ, at around 3.0 and 4.0 are chosen by the algorithm. Discarding the sample variance of the histograms and replacing it with variance at 4.0 clearly increases the accuracy, which can be interpreted as the search space defocuses artificially to have a broader knowledge about the neighborhood.

Comparing the optimization strategy with a previous work [23], we see that using a search strategy is more advantageous in terms of accuracy than a simple grid search. A grid search remains superficial, but the MCPSO can focus on and make in-depth searches in the parameter space. This leads to lower error values and higher accuracies for positioning purposes. The preliminary experimental results show that higher accuracies are highly possible by choosing higher decay factors with narrower parameter values.

3.3. Analysis of the State Vector Values

The errors reported in the previous section are informative snapshots of the best parameters and the best accuracies achieved so far. However, in the search for generalized parameters, the optimization process also enables us to analyze the evolution of the individual parameters, so that intervals or even joint effects of the parameters can be analyzed. For this section, we select the trajectory $\mathcal{E}_3$ to do far more detailed experiments.

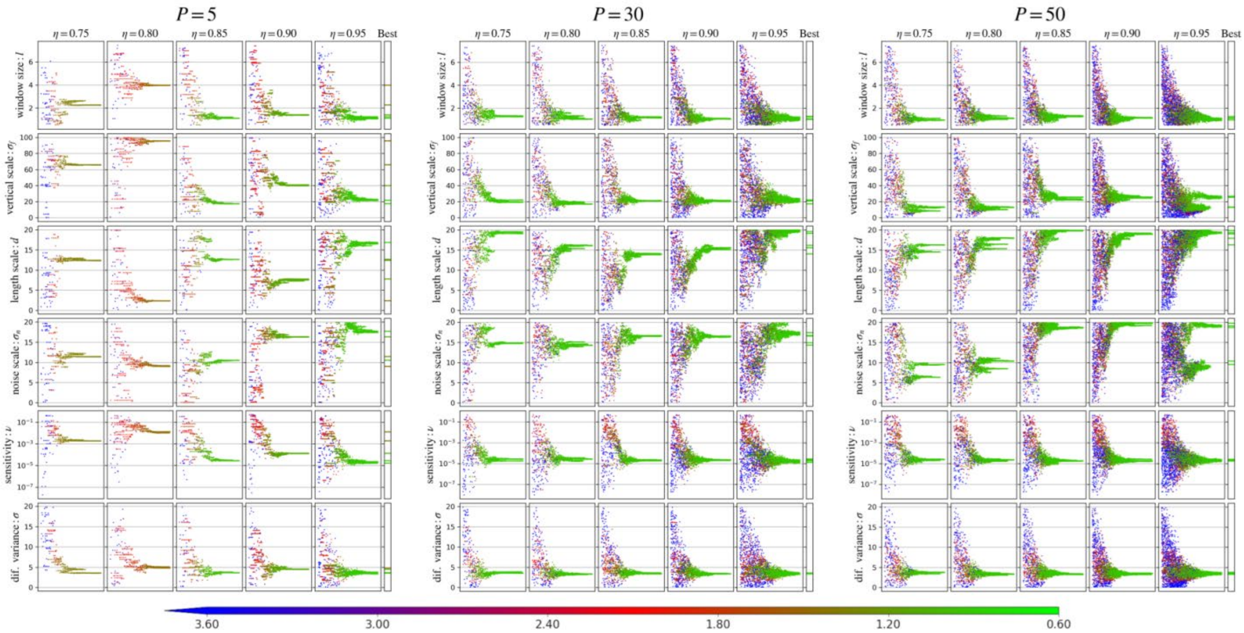


Figure 4. Evolution of the state vector through the optimization process with respect to different decay factors (η) and population sizes (P).

In the optimization process, the values of the state vector are initialized uniformly between the extremities, and we expect them to converge to the optimum values. In Figure 4, we visualize the evolution of the individual parameters of the state vector through the optimization process of the experiment $\mathcal{E}_3$. We analyze the process from the perspective of three different particle population sizes ($P \in \{5,30,50\}$) and with respect to different decay factors ($\eta \in \{0.75,0.80,0.85,0.90,0.95\}$). For each case, we visualize the value of a parameter as a dot whose color is an indicator for the error median of the positioning performance related to that parameter. As the color bar suggests, towards the bluish part of the color scale, the responses represent high positioning errors, and the greener colors are the responses with lower errors. Because the initial particle population is sampled from a uniform distribution, we see a gradient of colors from blue, red and green through the process. We also mark the converged values with the colors they have at the end in the column “Best”.

In the optimization process, the values of the state vector are initialized uniformly between the extremities, and we expect them to converge to the optimum values. In Figure 4, we visualize the evolution of the individual parameters of the state vector through the optimization process of the experiment $\mathcal{E}_3$. We analyze the process from the perspective of three different particle population sizes ($P \in \{5,30,50\}$) and with respect to different decay factors ($\eta \in \{0.75,0.80,0.85,0.90,0.95\}$). For each case, we visualize the value of a parameter as a dot whose color is an indicator for the error median of the positioning performance related to that parameter. As the color bar suggests, towards the bluish part of the color scale, the responses represent high positioning errors, and the greener colors are the responses with lower errors. Because the initial particle population is sampled from a uniform distribution, we see a gradient of colors from blue, red and green through the process. We also mark the converged values with the colors they have at the end in the column “Best”.

We further observe the convergence of the scale factors, the signal variance (σ_f) and the noise variance (σ_n). In Figure 5, the convergence set of these parameters, or in other words, the values of the population at the end of the optimization process, is plotted. With the high population sizes $P = 30$ and $P = 50$ as being more reliable, we disregard the values of the lower population sizes. The resulting points are distributed around a line as shown in the graph, which shows that there is a linear relation between these two parameters, and the values converge to a line instead of individual values.

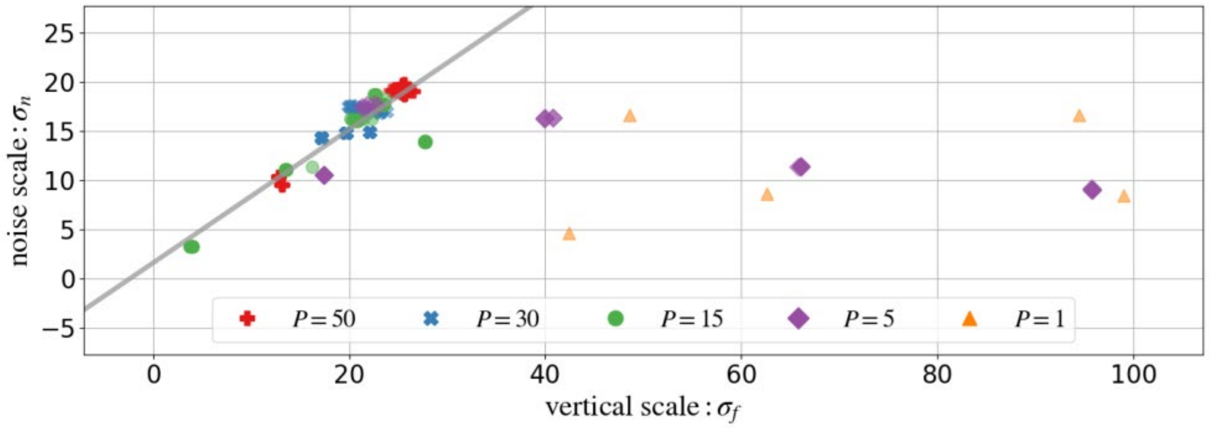


Figure 5. Line fit on the converged values of the two GPR parameters: the signal variance and the noise variance.

3.4. Population and Time Performance Analysis

The system is composed of heavy processes in terms of computation and storage. We hypothesize that lower error values can be achieved with very limited swarm population size and number of iterations, with a proper discretization strategy. To limit the number of iterations, we tune the decay factor: A Lower decay factor leads to faster convergence. In Figure 6, various configurations of the population size and the decay factor are shown for a single experimental trajectory, $\mathcal{E}_1$. We see that we achieve a median error value of 0.77 m for a decay factor of 0.95 and a population size of 50. Lower population sizes lead to considerably higher error values. The error statistics fall as the size of the swarm population is increased, that is, by evaluating considerably greater numbers of state vectors at the initial iterations.

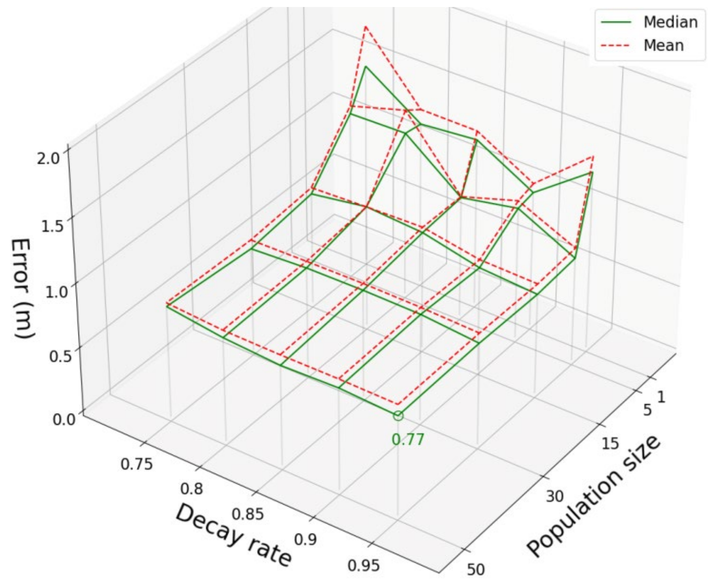


Figure 6. Accuracy comparison of the MCPSO with respect to the population size and the decay factor.

We further investigate the influence of the particle size on the convergence duration and the error performance. We show the results of the experiments (on $\mathcal{E}_1$) with 5, 30 and 50 particles, in Figure 7. The repetition of the stages can be clearly distinguished in all the figures. The duration falls gradually down to 0 seconds when stage repetitions are available, especially after the 30th iteration with decay factors of 0.75 and 0.85. Expectedly, the experiments with 50 particles perform better in terms of accuracy for all decay factors, but their time performances are the worst, with each iteration taking about 4000 seconds, only to generate an error statistic. The durations fall with the number of particles per population; however, the experiments with a lower number of particles perform comparably with the 50 particles at highly lower durations. The difference between the medians is of the order of 0.03 meters. Considering this small error difference, lower particle sizes with higher decay rates are favorable. We also note that the choice of the decay rate affects the performance. A decay rate should be compatible with the population size. The experiment with a rate of 0.75 and 5 particles is clearly incomplete. However, selecting a high decay rate for 5 particles performs almost as well as selecting 50 particles. To summarize, if the purpose is to decrease the positioning error, we choose a high decay rate. If we have further considerations on the duration of the experiments, we will choose lower particle sizes.

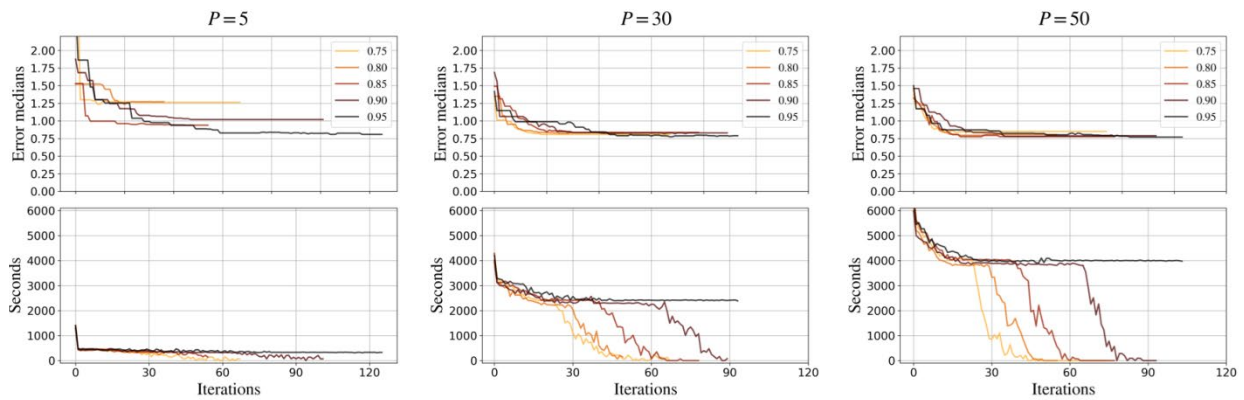


Figure 7. Accuracy and duration comparison of the iterations with respect to the decay factor (η) and population size (P).

3.5. Fine-tuned Experiments and Results

Considering the issues of analysis of the converged values, we reboot the experiments using more condensed intervals for the window size ($\ell \in [0.5,0.3]$), the sensitivity ($\nu \in [10^{-7}, 10^{-2}]$, the diffusion variance ($\sigma \in [2.5,5.0]$). The previous analysis shows that the dependency between the GPR parameters is more important than their ranges. So, the intervals should be selected by allowing them to move freely in the dependency ranges. We modify the search space of the GPR parameters as $\sigma_f \in [0.1,20000]$, $d \in [0.1,200]$, and $\sigma_n \in [0.1,20000]$. Moreover, the new experiments are repeated by setting the population size to $P = 30$ and the decay rate to $\eta = 0.90$, as these values are shown to perform similarly to their higher-resolution versions (see Figure 6).

We report the final errors in Table 3. We achieve significantly lower errors for individual experiments and comparable errors for the whole experiment. We deduce that the performance of the IPS can be increased by fine-tuning the parameters using an efficient optimization procedure.

Table 3. Best error statistics (m) and the corresponding parameter configuration for an L_1-norm kernel after the secondary experiments.

Exp.	ℓ	σ_f	d	σ_n	ν	σ	Median	Mean
$\mathcal{E}_1$	1.4	84.5	4.05	13.27	0.0001	4.26	0.979	1.114
$\mathcal{E}_2$	3.19	79.0	4.02	7.67	0.004	3.27	1.068	1.289
$\mathcal{E}_3$	1.3	85.1	12.89	8.73	0.0002	3.29	0.704	0.827
$\mathcal{E}_4$	2.05	78.3	3.51	19.42	0.002	3.75	0.846	1.646
All	2.72	48.3	5.14	9.66	0.0005	4.5	1.107	1.42

3.6. Discussion

With the final results, we show that it is possible to increase the accuracy of a BLE fingerprint-based IPS system if the whole system can be redefined with an optimization approach. Keen readers can understand that all patterns in the dataset are used to obtain the convergence parameters and the best input parameter configuration, and the system can be converted into a parameter learning approach with well-known automated learning techniques. However, there are three fundamental problems that prevent us from using standard validation techniques. The first one is that we lack a dataset in which the input and output parameter values are previously determined. We only have access to the RSSI data and their corresponding ground-truth position. An IPS system should be run as a whole to link the RSSI data to the positions. Secondly, the parameters are not independent of each other. The input of each box can also affect the output of a subsequent stage, for which we added a section for such an analysis. And thirdly and most importantly, being also related to the first of these problems, a single evaluation of a parameter configuration takes enormous processing power and time to move to a pure learning setup, which is inevitably intractable in the long run.

4. Conclusion

This work introduces an optimization strategy that is used for fingerprinting-based indoor positioning purposes. RSSI data collected on some RPs in an indoor area of interest are exposed to a series of stages: a maxfilter step that filters the most beneficial information in the RSSI data, a GP-based RM estimation method that constructs a likelihood of RSSI data with respect to a position, and an adaptive particle filter based trajectory estimation that yields a list of positions the object of interest travels upon. With the help of ground-truth positions, we can assess the final trajectory estimations, hence the performance of the parameters of the stages. This leads us to close the loop and turn the problem into an optimization problem that can be solved by a specially designed Monte Carlo particle swarm optimization strategy. The special design is a custom discretization of the parameter space that allows repetitions of the parameters, so that the preceding heavy processes are not always required to run. We show that the search is performed more intelligently, focusing on the parameter regions with low errors, and the search time is decreased drastically with respect to a grid-based search.

There are multiple ways to extend this work: (i) The very first way would be to replace the exponential kernel used in this study with custom kernels, Matèrn kernel, rational quadratic kernel or dot-product kernel. Furthermore, being able to evaluate the results, a deep learning strategy can be implemented to learn the RMs. (ii) The trajectory estimation can be performed by using other filtering techniques: a forward algorithm can be introduced by properly discretizing the area, or the Kalman filter and its extensions will be beneficial if the relation between the position and the RSSI data can be approximated with Gaussians. Each technique will bring its own parameters. By discretizing the parameter spaces as shown in the methodology, any appropriate technique can be plugged into the presented optimization system.

References

[1] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2568–2599, 2019.

[2] R. Brena, J. Garcia-Vázquez, C. Galván Tejada, D. Muñoz, C. Vargas-Rosales, J. Fangmeyer Jr, and A. Palma, "Evolution of indoor positioning technologies: A survey," Journal of Sensors, vol. 2017, 03 2017.

[3] T. Arsan, "Accurate indoor positioning with ultra-wide band sensors," Turkish Journal of Electrical Engineering and Computer Sciences, vol. 28, no. 2, pp. 1014–1029, 2020.

[4] M. Liu, L. Cheng, K. Qian, J. Wang, J. Wang, and Y. Liu, "Indoor acoustic localization: a survey," Hum.-Centric Comput. Inf. Sci., vol. 10, Jan. 2020.

[5] D. Plikynas, A. Žvironas, A. Budrionis, and M. Gudauskis, "Indoor navigation systems for visually impaired persons: Mapping the features of existing technologies to user needs," Sensors, vol. 20, no. 3, 2020.

[6] A. De-La-Llana-Calvo, J.-L. Lázaro-Galilea, A. Gardel-Vicente, D. Rodriguez-Navarro, I. Bravo-Muñoz, and F. Espinosa-Zapata, "Characterization of multipath effects in indoor positioning systems by aoa and poa based on optical signals," Sensors, vol. 19, no. 4, 2019.

[7] A. Pérez-Navarro, J. Torres-Sospedra, R. Montoliu, J. Conesa, R. Berkvens, G. Caso, C. Costa, N. Dorigatti, N. Hernández, S. Knauth, E. S. Lohan, J. Machaj, A. Moreira, and P. Wilk, "Challenges of fingerprinting in indoor positioning and navigation," in Geographical and Fingerprinting Data to Create Systems for Indoor Positioning and Indoor/Outdoor Navigation (J. Conesa, A. Pérez-Navarro, J. Torres-Sospedra, and R. Montoliu, eds.), Intelligent Data-Centric Systems, pp. 1–20, Academic Press, 2019.

[8] S. Subedi and J.-Y. Pyun, "Practical fingerprinting localization for indoor positioning system by using beacons," Journal of Sensors, vol. 2017, no. 1, p. 9742170, 2017.

[9] F. S. Daniş and A. T. Cemgil, "Model-based localization and tracking using bluetooth low-energy beacons," Sensors, vol. 17, no. 11, 2017.

[10] R. Shrestha, D. Romero, and S. P. Chepuri, "Spectrum surveying: Active radio map estimation with autonomous uavs," IEEE Transactions on Wireless Communications, vol. 22, no. 1, pp. 627–641, 2023.

[11] M. Nabati and S. A. Ghorashi, "A real-time fingerprint-based indoor positioning using deep learning and preceding states," Expert Systems with Applications, vol. 213, p. 118889, 2023.

[12] F. Alhomayani and M. H. Mahoor, "Deep learning methods for fingerprint-based indoor positioning: a review," Journal of Location Based Services, vol. 14, no. 3, pp. 129–200, 2020.

[13] Y. Zhao, C. Liu, L. S. Mihaylova, and F. Gunnarsson, "Gaussian processes for rss fingerprints construction in indoor localization," in 2018 21st International Conference on Information Fusion (FUSION), pp. 1377–1384, 2018.

[14] R. Guan, A. Zhang, M. Li, and Y. Wang, "Measuring uncertainty in signal fingerprinting with gaussian processes going deep," in 2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 1–8, 2021.

[15] F. S. Daniş, "Live RSSI filtering for indoor positioning with bluetooth low-energy," in 2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 1–8, 2022.

[16] F. S. Daniş, A. T. Cemgil, and C. Ersoy, "Adaptive sequential monte carlo filter for indoor positioning and tracking with bluetooth low energy beacons," IEEE Access, vol. 9, pp. 37022–37038, 2021.

[17] F. S. Daniş, A. T. Naskali, A. T. Cemgil, and C. Ersoy, "An indoor localization dataset and data collection framework with high precision position annotation," Pervasive and Mobile Computing, vol. 81, p. 101554, 2022.

[18] D. Luengo, L. Martino, M. Bugallo, V. Elvira, and S. Särkkä, "A survey of monte carlo methods for parameter estimation," EURASIP Journal on Advances in Signal Processing, vol. 2020, may 2020.

[19] O. Akyildiz, D. Crisan, and J. Miguez, "Parallel sequential monte carlo for stochastic gradient-free nonconvex optimization," Statistics and Computing, vol. 30, 11 2020.

[20] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain, "Gaussian processes for time-series modelling," Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 371, no. 1984, p. 20110550, 2013.

[21] C. E. Rasmussen, Gaussian Processes in Machine Learning, pp. 63–71. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[22] D. Barber, Bayesian Reasoning and Machine Learning. USA: Cambridge University Press, 2012.

[23] F. S. Daniş, "Practical and parameterized fingerprinting through maximal filtering for indoor positioning," IEEE Journal of Indoor and Seamless Positioning and Navigation, vol. 1, pp. 199–210, 2023.

Article Information Form

Support/Supporting Organizations

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Ethical Approval

It is declared that during the preparation process of this study, scientific and ethical principles were followed, and all the studies benefited from are stated in the bibliography.

Availability of data and material

The data is available with the DOI 10.34740/kaggle/ds/1662453.

Artificial Intelligence Statement

No artificial intelligence tools were used while writing this article.

Plagiarism Statement

This article has been scanned by iThenticate™.