# Towards Feasible Wi-Fi based Indoor Tracking Systems Using Probabilistic Methods

Lorenz Schauer, Philipp Marcus, and Claudia Linnhoff-Popien
Mobile and Distributed Systems Group
Ludwig-Maximilians-Universität München
Munich, Germany
Email: lorenz.schauer@ifi.lmu.de, philipp.marcus@ifi.lmu.de, linnhoff@ifi.lmu.de

***Abstract*—Wi-Fi enabled devices periodically broadcast unencrypted management information which can easily be used for an involuntary tracking of users in an area of interest. However, reliable trajectory estimations on this data remain challenging, due to arbitrary and imprecise position fixes of moving targets. Probabilistic methods can help to increase the estimation accuracy significantly, but may degrade other important metrics, e.g. scalability, complexity, or robustness. In this paper, we investigate probabilistic solutions for a feasible tracking system for indoor scenarios. Beside the usage of Viterbi's algorithm and a common particle filter, we propose a novel state particle filter with a more restricted transition model based on discrete state nodes. All methods are compared and evaluated on various user traces using real Wi-Fi captures from common mobile devices at our office building. The results indicate that the proposed state particle filter performs best in terms of accuracy, and precision while using a smaller amount of particles which renders this approach scalable, and thus, feasible for indoor tracking systems.**

## I. Introduction

For an ubiquitous network connectivity, Wi-Fi enabled devices automatically scan for access points (APs) in their vicinity sending out IEEE 802.11 probe requests. These unencrypted frames contain certain management information, e.g., the device's specific MAC address, or distinct network names. Such information can be easily captured by any Wi-Fi card in reach operating in monitor mode. This technique has been successfully used in order to estimate pedestrian flows [1], or user trajectories [2]. Overall, capturing Wi-Fi signals from passing mobile devices involve high potentials for a low-cost and fine-grained indoor pedestrian tracking system, due to the fact, that no additional hardware, nor any active user participation is required. Furthermore, this technique renders current estimations of actual user flows possible being of great interest for management, commercial or emergency purposes in public buildings. Beside these potentials, however, the reliability of such Wi-Fi based tracking systems is often limited, due to imprecise position fixes of moving targets, as demonstrated in this work. Wi-Fi location estimates are often inaccurate, due to physical factors, e.g., fluctuations in received signal strengths caused by multipath propagation effects.

In this paper, we address these limitations, in order to make Wi-Fi based indoor tracking systems more accurate and feasible. Therefore, we use knowledge about humans' mobility and investigate well-adopted probabilistic methods. More specific, we apply the Viterbi algorithm, and a commonly used sequential importance resampling (SIR) particle filter on a weighted kNN fingerprinting technique based on passive Wi-Fi captures. For this purpose, we take advantage of a Hidden Markov Model (HMM) based on discrete state nodes which cover all possible paths a user can take within a given building.

Furthermore, we introduce another particle filter with a more restricted transition model, related to the coordinates of the proposed state nodes of our HMM. The aim of our so-called state particle filter (sPF) is to achieve adequate results in terms of accuracy and robustness, while reducing the amount of particles for scalability reasons. To our knowledge, this has not been presented for Wi-Fi based indoor tracking systems.

All methods are evaluated on various user traces based on real Wi-Fi captures from mobile devices at our office building. They are compared in terms of common metrics for indoor positioning systems, e.g., accuracy, precision, etc. Furthermore, the impact of the state node density is investigated, and a heuristic for an adequate state node placement is introduced.

The reminder of this paper is structured as follows: Section II briefly reveals related work. Section III describes technical background and presents basic principles of our methodologies. Our evaluation is introduced in Section IV, and finally, Section V concludes this paper giving hints on future work.

## II. Related Work

Wi-Fi based indoor tracking involves a long research history. RADAR [3], is seen as one of the first in-building tracking systems. It uses Wi-Fi fingerprinting with a nearest neighbor algorithm achieving a median spatial error distance of 2.94 meters. In an enhanced version, Bahl et al. performed a Viterbi-like algorithm for continuous user tracking and reduced the error distance to 2.37 meters [4]. Krumm and Horvitz [5] applied two HMMs and inferred motion and location information of a mobile client based on Wi-Fi measurements. Like us, they used Viterbi on a graph of discrete location nodes, and a similar transition model respecting pedestrians' speeds. They obtained a median error of 1.53 meters based on ten short walks while performing Wi-Fi active scans.

In recent years, real-time tracking has gained high attention, due to the increased usage of mobile devices. Trogh et al. [6] presented a real-time indoor tracking system using Viterbi and semantic data. Position fixes are performed similarly to our work, using Wi-Fi fingerprinting on a pre-calculated radio map on grid points. An average median accuracy of below 2 meters has been obtained. Musa and Eriksson [2] performed passive Wi-Fi scans at different monitors for tracking unmodified smartphones. They estimated the most likely trajectories of individuals for outdoor scenarios using Viterbi algorithm on a HMM at road networks. They observed a mean error under 70 meters compared to GPS readings. Other Wi-Fi based indoor tracking systems calculate the location probability distribution of a mobile target by considering the difference in the received signal strengths at two monitors [7], or perform a lateration algorithm achieving a lower mean accuracy of 15 meters [8].

Beside the Viterbi algorithm, particle filters combined with floor plan information are also widely adopted for probabilistic localization and tracking [9], [10]. Bartoletti et al. [11] present a passive indoor tracking system of moving targets using a common particle filter and time of arrival estimations at ultra-wideband sensors. Khan et al. [12] compared Kalman filter, extended Kalman filter, and particle filter for target tracking using signal strength measurements. Chen et al. [13] investigated multiple target tracking for indoor scenarios, and discussed simplifications to the SIR particle filter.

However, none of the presented works compare performance metrics of both Viterbi algorithm and particle filter for Wi-Fi based indoor tracking systems, and thus, we fill in this gap. Furthermore, a state particle filter focusing on state node locations of an underlying HMM, as it is proposed in this work, could not have been found in literature.

## III. Basic Principles

In this section, we firstly give a brief overview of technical background for passive Wi-Fi tracking, before presenting basic principles of our methods.

### A. Wi-Fi Tracking

From a mobile device's perspective, network discovery can be performed passively, or actively. For the first, a client merely listens on beacon frames which are periodically transmitted by APs over all operating channels. Due to mismatching channels, clients may miss transmitted beacons using the passive procedure. Therefore, mobile devices prefer active scanning, due to lower energy-consumption and shorter discovery time of APs [14]. In this case, clients send out probe requests iteratively for each channel and wait for a certain period while listening for corresponding probe responses.

Probe requests contain unencrypted device specific information, such as the client's MAC address, and other management information. These frames can be sniffed by any Wi-Fi card operating in monitor mode within communication range. Hence, continuous active scans captured by different monitor nodes in an area of interest can be used to track users' locations and corresponding trajectories, as proven in [2]. However, the wide communication range, the fluctuations in received signal strengths (RSS), and the arbitrary bursts of probe requests transmitted by mobile devices render position fixes inaccurate. Hence, accurate indoor positioning techniques are required for our purpose.

### B. Passive Wi-Fi Localization

Wi-Fi fingerprinting, firstly presented in RADAR [3], is one of the most popular technique for indoor localization. Basically, it consists of two phases: an offline training phase where an RSS vector of all APs in reach is determined at certain reference points. The vectors and corresponding locations are stored into a radio map. In an online phase, the user actively scans for APs in reach, and the resulting RSS vector is compared with the entries in the radio map. Finally, the most probable user location is returned. Note, that in case of passive Wi-Fi localization, the mobile phone scans its vicinity without an active user participation, and the RSS vector is determined by the monitor nodes. Wi-Fi fingerprinting achieves adequate positioning results, but also requires high efforts for recording the radio map. To bypass this problem, we determine the radio map by calculating the RSS vector for each reference point to all available APs using the standard logarithmic path loss model with wall attenuation factor (WAF) for indoor environments. Related to [15], the received signal power from a certain reference point at an AP is calculated by the following equation:

$$P_{rx}(d) = A - 10n\log_{10}(d) - l \cdot \mathrm{WAF} \tag{1}$$

with $A$ being the transmission power, $n$ denotes the path loss exponent, and $l$ is the amount of walls arranged in the Euclidean distance $d$ between sender and receiver. Note, that the model parameters $n$, $A$, and WAF are empirically determined for our test set.

A position fix for a mobile device with MAC address MAC at time $t$ is performed, when an RSS observation $\mathrm{obs}_{\mathrm{MAC},t,m}$ is made at a Wi-Fi monitor node $m \in M$. Due to the nature of probe request burst, $\mathrm{obs}_{\mathrm{MAC},t,m}$ is calculated as the mean of captured RSS values from one MAC address at $m$ during the time span $[t, t + \Delta t]$, rather than considering only a point in time. The RSS observation vector $\overrightarrow{v_o}$ is determined based on the observations of all monitor nodes: $\overrightarrow{v_o} = (\mathrm{obs}_{\mathrm{MAC},t,m_1}, ..., \mathrm{obs}_{\mathrm{MAC},t,m_M})$. If there is no observation made at $m$ for MAC during the time span $[t, t+\Delta t]$, the corresponding entry is set to $\mathrm{obs}_{\mathrm{MAC},t,m} = -100$ dBm, as it is proposed in [16] for treatment of missing RSS values.

Based on the calculated radio map and $\overrightarrow{v_o}$ of an observed target, we perform a common deterministic location estimation using weighted kNN classification, with $w_i$, $i \in \{1, ..., k\}$ being determined by:

$$w_i = \left( d_i \sum_{j=1}^{k} \frac{1}{d_j} \right)^{-1} \tag{2}$$

where $d_i = dist(\overrightarrow{v_o}, \overrightarrow{r_i})$ denotes the Euclidean distance in signal space between $\overrightarrow{v_o}$ and the specific record $\overrightarrow{r_i}$ of the radio map. The target's position is then estimated by the sum of weighted $k$ position candidates. Considering the time series of passive Wi-Fi localizations, we are interested in the most accurate path taken by the user. Therefore, we consider both Viterbi algorithm and particle filter, as well-adopted probabilistic methods, and map them to our problem.

### C. Viterbi Algorithm

The Viterbi algorithm provides an optimal solution to the problem of estimating the most likely sequence of a discrete time finite-state Markov process [17]. Hence, for our trajectory estimation problem, we firstly define a HMM $\lambda = (S; O; A; B; \pi)$ with hidden states $S$, observations $O$, transition probabilities $A$, observation probabilities $B$, and an initial probability distribution $\pi$.

*1) Hidden States:* The state model must cover all paths a user can take in a fine-granular manner. Thus, we deploy state nodes based on both the building topology and pedestrians' mobility. Indoor environments typically consist of rooms, and corridors. We focus on the latter, due to the assumption, that passive Wi-Fi based estimations do not provide sufficient accuracy for tracking on room level, and generally, people mainly move within corridors. According to the map-matching problem on road networks in [2], we treat corridor intersections as vertices and corridors as edges which are divided into smaller segments for a higher granularity. Each vertex and segment is represented by one state node.

In contrast to road networks, a person can stay, move on, or return at each state node without any turn-restrictions. Consecutive state nodes are placed in line-of-sight condition with a certain distance $d_{i,j}$ between node $i \in S$ and node $j \in S$ which we determine using the following heuristic: considering an average pedestrian speed $v_p$ and the time span of Wi-Fi observations $\Delta t$, the distance is calculated as $d_{i,j} = v_p \cdot \Delta t$. This ensures, that state transitions in our HMM for every Wi-Fi observation also return the most probable state node for a person moving on average speed, and at least one state node during an observation time span is passed. Notice, Viterbi expects periodic observations which may also include no detections for a specific MAC address at time $t$.

*2) Transition Probabilities:* A transition is performed based on periodic observations and the underling transition probabilities. If no observation is made for a MAC address during one step, we cannot make any assumption about the target's movement. If an observation is made and the RSS vector $\overrightarrow{v_o}$ is determined, we calculate the time difference $\delta t$ to the last observation of this target. Considering the distances from the actual state node $i$ to any other state node $x$ in the given building, we determine the velocity $v_x$ required for reaching $x$ in the given time by calculating $v_x = \frac{d_{i,x}}{\delta t}$, with $d_{i,x}$ being the walking distance from $i$ to $x$. Assuming, that the pedestrian's velocity is a normal distributed random variable $V \sim \mathcal{N}(\mu, \sigma^2)$ with $\mu = v_p$ and $\sigma^2 = 1$, we consider a folded normal distribution $Y = |V|$ and test $v_x$ against the probability density function (pdf) $f_Y$, in order to determine the transition probability $p(i \rightarrow x)$ from $i$ to $x$:

$$p(i \rightarrow x) = f_Y(v_x; \mu, \sigma^2) \tag{3}$$

As initial probability distribution $\pi$, we consider the observation probability over the first observation.

*3) Observation Probabilities:* As mentioned before, without observations we have no information about the person's movement. This is contrary to [2], where non-detections are considered using an underlying emission probability model for mobile phones. However, dealing with non-detections is critical for us, due to two reasons: first, a generic emission probability model for the numerous types of existing mobile devices is hard to find, and second, due to the arbitrary nature of probe request bursts, non-detections do not implicitly prove that a target has not passed a monitor's coverage range. Hence, we consider observations only and determine their probabilities for our HMM as follows: a theoretical RSS vector $\overrightarrow{v_s}$ is pre-calculated for each state node using the standard logarithmic path loss model from Section III-B. As in the Horus system [18], we assume that an RSS measure $X$ at a certain location from one transmitter follows a normal distribution $X \sim \mathcal{N}(\mu, \sigma^2)$, with mean $\mu$, and standard deviation $\sigma$. The observation probability $p(\text{obs} \,|s)$ at a state node $s \in S$ for a given observation $\overrightarrow{v_o}$ is determined using the pdf $f_X$:

$$p(\text{obs} \,|s) = \prod_{i=1}^{M} f_X(o_i; \mu_i, \sigma^2) \tag{4}$$

with $o_i \in \overrightarrow{v_o}$, $\mu_i \in \overrightarrow{v_s}$, and $M$ being the set of Wi-Fi monitors.

*4) Viterbi-Path Estimation:* We have just described our HMM $\lambda = (S; O; A; B; \pi)$. Given $\lambda$, and a continuous time-series of observations for a particular MAC address $o = \overrightarrow{v_{o1}}, ..., \overrightarrow{v_{oT}} \in O^*$, the Viteri algorithm produces the most likely sequence of hidden states $q^* = s_1^*, ..., s_T^* \in S^T$, called Viterbi-Path which denotes the most probable user trajectory. However, fix hidden state nodes restrict the solution space, and may negatively influence the estimation accuracy. Therefore, we now investigate a particle filter technique.

### D. Particle Filter

Particle filters present a nonparametric alternative to the Bayes filter dealing with non-linear and non-Gaussian estimation problems. The main idea is to represent the posterior believe by a set of random state samples (called particles) with associated weights, in order to compute estimates based on this believe. The basic filter algorithm, known as sequential importance sampling (SIS), approximates the posterior density $p(X_k|Z_k)$ of states $X_k$, considering the past observations $Z_k$ up to time $k$ [19]:

$$p(X_k|Z_k) \approx \sum_{i=1}^{N} w_k^i \, \delta(X_k - X_k^i) \tag{5}$$

where $X_k^i$ denotes the $i$-th particles up to time $k$ of the posterior with weight $w_k^i$. One key feature is the resampling step, where particles are re-sampled according to their weight. A so-called sequential importance resampling (SIR) filter is also applied in this work consisting of the following steps:

*1) Initialization step:* For initialization, a temporary set of $N$ particles $\overline{X}_0 = \{x_0^1, ..., x_0^N\}$ is randomly placed on free space of the building plan for time 0 using a continuous uniform pdf $f(x_0^i)$. Each particle $x_0^i$ is then located at $l_0^i$. It has an assigned initial velocity $v_0^i$ with orientation $o_0^i$ based on the continuous uniform pdf from 0 to $v_{\max}$, and 0 to $2*\pi$, respectively. We consider $v_{\max} = 10\,ms^{-1}$ as maximum pedestrian's speed for indoor scenarios.

*2) Prediction Step:* For each particle $x_k^i \in \overline{X}_k$, we determine a new particle $x_{k+1}^i$ with location $l_{k+1}^i$, its velocity $v_{k+1}^i$, and its new orientation $o_{k+1}^i$ using the transition pdf $p(x_{k+1}^i | x_k^i)$ based on the underlying mobility model according to Widyawan et al. [20]:

$$l_{k+1}^i = \begin{bmatrix} l_{k+1_x}^i \\ l_{k+1_y}^i \end{bmatrix} = \begin{bmatrix} l_{k_x}^i + v_k^i \cos(o_k^i)\Delta t + n_k \\ l_{k_y}^i + v_k^i \sin(o_k^i)\Delta t + n_k \end{bmatrix} \tag{6}$$

As it is shown, a particle's new position is derived from its last position, velocity, and orientation considering the elapsed time $\Delta t$ to the last prediction step, and an optional noise parameter $n_k$. In our case, $n_k$ is ignored, and $\Delta t$ depends on both the timestamp of the last Wi-Fi measurement and the humans' step duration which is treated as a constant parameter $c_{\text{step}} = 0.8\,s$. The new velocity and orientation is calculated for each prediction step using the following equations:

$$v_{k+1}^i = |\mathcal{N}(v_k^i, 1\,ms^{-1}\Delta t)| \tag{7}$$

$$o_{k+1}^i = \mathcal{N}\left(o_k^i, 2\pi - \arctan\left(0.5\sqrt{v_k^i}\,\Delta t\right)\right) \tag{8}$$

with $\mathcal{N}(\mu, \sigma^2)$ denoting a random choice out of a normal distribution with mean $\mu$ and standard deviation $\sigma$.

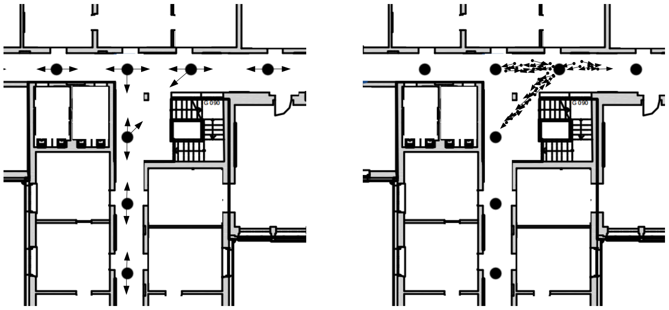
*3) Importance Sampling Step:* In this step, the measurement $z_{k+1}$ is incorporated into the new particle set by calculating the importance factor, denoted as a individual particle weight $w_{k+1}^i$. This reflects the probability of $z_{k+1}$ under the particle $x_{k+1}^i$, denoted as $w_{k+1}^i = p(z_{k+1} | x_{k+1}^i)$. Note, that the weight of particles which have crossed a non-walkable space, is set to 0. Overall, the set of weighted particles approximate the Bayes filter posterior believe.

In our case, a measurement $z_k$ consists of an observed RSS vector $\overrightarrow{v_o}$. In order to calculate the particles' weight, we determine the measurement distribution $(\mu, \sigma)$, with $\mu$ being the position fix returned by the proposed location estimation technique of Section III-B. For determination of $\sigma$, we use the Laplace error estimator, introduced by Marcus [21]. Finally, a particle's weight can be directly derived from the pdf of the Laplace error estimator using its current location $l_{k+1}^i$:

$$p(z_{k+1} | x_{k+1}^i) = \text{pdf}_{\mu,\sigma}^{\text{laplace}}(l_{k+1}^i) \tag{9}$$

As last step, the particle weights are normalized resulting in a total weight sum of $\sum_{i=1}^N w_{k+1}^i = 1$.

*4) Resampling Step:* This is a key step of SIR particle filters, where low-weighted particles are probably destroyed, and high-weighted particles are duplicated. This is performed by drawing $N$ particles with replacement out of the current particle set according to the associated weights. The result is a new temporary set of $N$ particles $\overline{X}_{k+1}$ which is given to the next prediction step.


(a) Initialization Step

(b) Prediction Step

Fig. 1. Illustration of state particle filter where particles move according to their orientation (marked as arrows) based on state node locations (black dots).

*5) Path Estimation:* For each iteration, a person's position is determined by the average location of $\overline{X}_{k+1}$. The resulting position sequence after all iterations represents the most probable user trajectory. Due to the sequential processing for a finite set of $N$ particles, we obtain a linear time complexity of $O(N)$. Hence, the proposed SIR particle filter indicates a higher scalability than Viterbi while $N \ll |S|^2$. However, conducted experiments, e.g., in [9], or [10], show that 400 and more particles are required to achieve an adequate accuracy minimizing the scalability. Therefore, a modification to the described algorithm is now introduced.

### *E. State Particle Filter*

The goal of our state particle filter (sPF) is to achieve accurate estimations while keeping the amount of required particles low for scalability reasons. For this purpose, we combine the fixed hidden state model of Section III-C, with the open space mobility model of Widyawan et al. [20]. Hence, we apply the following restrictions to the initialization and prediction step of the described SIR particle filter:

*1) Initialization Step:* Instead of placing particles randomly, we now consider the locations of $|S|$ hidden states and place a set of $N$ particles uniformly at these locations, with $N = i \times |S|$, $i \in \mathbb{N}$. Again, each particle assigns an initial velocity $v_0^i$ and orientation $o_0^i$. While $v_0^i$ is still determined by the continuous uniform pdf like before, $o_0^i$ is now randomly set to one of the visible neighbor state nodes, as exemplarily depicted in Figure 1a. Note, that a state node is visible if it's in line-of-sight condition to the particle. Overall, particles are initially forced to move along the state node graph.

*2) Prediction Step:* In order to map the transition pdf $p(x_{k+1}^i | x_k^i)$ to our state node model, a particle's orientation change is restricted by the state node locations. Instead of using Equation 8, a particle's orientation $o_k^i$ is now set according to the angle between its current location $l_k^i$ and one randomly chosen visible state node. If more states are in line-of-sight to the particle within a small angle range $\alpha \pm 2.5°$, the particle will move towards the closest of these nodes. If it passes a certain state node during the prediction duration, the particle's

```
Require: $A$:= Angel matrix, $D$:= Distance matrix
  for all $i \in N$ do
    Determine $v_k^i$ from $v_{k-1}^i$ by Equation 7
    idx $= randomChoiceOf(isFinite(A[l_k^i]))$
    Set orientation: $o_k^i = A[l_k^i]$[idx]
    Determine distance to node: $dist = D[l_k^i]$[idx]
    Calculate time to node: $t_n = dist/v_k^i$
    while $t_n < \Delta t$ do
      $\Delta t = \Delta t - t_n$
      Set $l_{k+1}^i$ to location of selected node
      idx $= randomChoiceOf(isFinite(A[l_k^i]))$
      Set orientation: $o_k^i = A[l_k^i]$[idx]
      Determine distance to node: $dist = D[l_k^i]$[idx]
      Calculate time to node: $t_n = dist/v_k^i$
    end while
    Determine $l_{k+1}^i$ according to Equation 6
  end for
```

Fig. 2. Prediction step of the state particle filter in pseudocode.

location is set to this node and its orientation changes again towards another visible neighbor state node before its location is predicted. The effect is, that all particles move towards a certain state node's position at any time, and thus, their motion is based on the underlying state node graph, as exemplarily illustrated in Figure 1b.

The described prediction process is given as pseudocode in Figure 2. As you can see, we use two matrices $A$ and $D$ containing angel and distance information for every possible location to all visible state nodes. This information is easily accessible by using the corresponding indexes, as shown in the pseudocode. Furthermore, both matrices are precomputed for the whole setting, and thus, there is no overhead in comparison to other common particle filter implementations. The remaining steps for importance sampling and resampling are in accordance with the presented SIR particle filter. Hence, the proposed filter represents a mixed solution to the prior approaches where a fix state model according to Viterbi is combined with flexible particles of a SIR particle filter.

## IV. Evaluation

The proposed methods are now evaluated in terms of well-known performance metrics for indoor positioning systems, summarized by Liu et al. [22]. Therefore, we firstly present our real-world experiment in the sequel.

### A. Implementation and Setup

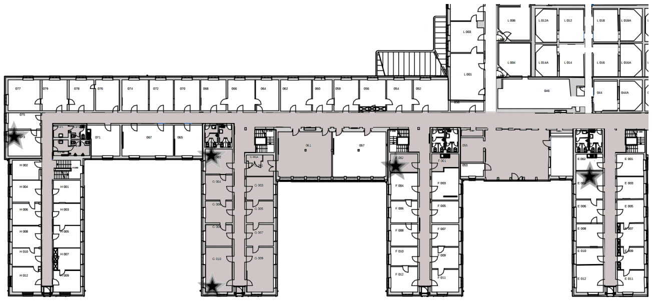


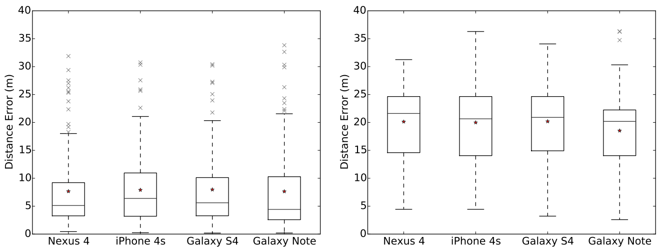
Fig. 3. Overview of experimental setup using 5 Wi-Fi monitors (black stars). Ground truth measurements are available within the gray marked region.

As depicted in Figure 3, we deploy five Wi-Fi monitors on the first floor of our office building covering a total area of 26.795 $m^2$. Each monitor is realized by a laptop equipped with a Netgear wireless PC card and Linux Debian 2.6. The monitors are online and synchronized by the network time protocol. All Wi-Fi adapters operate in monitor mode while running *tcpdum* for capturing probe requests. These captures are stored on a central server where relevant information is aggregated to one observation vector $\vec{v_o}$ for a particular MAC address and timespan $\Delta t$, as mentioned in Section III-B. We set $\Delta t = 5s$, due to a good trade-off between distinct RSS changes and preferably fine-granular position estimations. For map information, we use a bitmap representation of our office's building plan. The hidden states are deployed along all corridors, as described in Section III-C1. As default parameter, the average state node distance is about 6 meters, due to an assumed average pedestrian speed of $v_p = 1.2\,m/s$ and an observation time span $\Delta t = 5s$. For providing measurements with ground truth positions, we walk along all areas marked in gray in Figure 3, while performing Wi-Fi active scans at every 1.5 meters with four common smartphones, i.e., LG Nexus 4, iPhone 4s, Samsung Galaxy S4, and Samsung Galaxy Note. Overall, we collect Wi-Fi observations from 124 dedicated positions.

### B. Deterministic Estimation Accuracy

We firstly evaluate the estimation accuracy on the described data set without using probabilistic methods. Therefore, we determine the observation vectors for each test device and for each dedicated position, and calculate the location estimates, as described in Section III-B. Figure 4a indicates the error distributions in meters, considering the Euclidean distance between ground truth positions and estimations. It can be seen, that the test devices have a negligible influence on the captured signal strengths. Overall, we observe a root mean squared error of 7.37 meters on average which is not adequate for reliable indoor positioning systems. The interquartile ranges vary from 5.95 meters in case of the Nexus to 7.75 meters for the iPhone. The Galaxy Note shows the lowest median of 4.42 meters, but also the highest standard deviation of 7.50 meters. For all devices we observe extreme outliers containing a positioning error of up to 33.82 meters. In summary, these results indicate a low estimation accuracy.

Next, we investigate the trajectory estimation accuracy based on the observed position fixes. We create an observation graph over the ground truth positions, and generate 3 x 57 paths including 1, 2, and 3 destinations, while start point and destinations are randomly chosen. These artificial paths include our real Wi-Fi captures and represent the user trajectories. Given a time series of Wi-Fi observations, we now estimate the underlying trajectory by connecting the position



(a) Location Estimations (b) Trajectory Estimations

Fig. 4. Error distributions of our estimations to 124 ground truth locations and 171 trajectories using 4 different smartphones.

fixes base on these observation. Note, that each position fix is mapped to the state node graph by orthogonal projection, in order to be comparable to the subsequent methods.

For evaluation, we calculate the distance errors between the estimated trajectories and the corresponding paths by using the discrete Fréchet Distance as a well-adopted distance measure for trajectories. Figure 4b depicts the observed distance errors based on 171 artificial paths and the Wi-Fi observations from our 4 test devices. Again, it can be observed, that the different smartphones show a marginal variance of the estimation results obtaining a mean distance error of 18.52 meters for the Galaxy Note, and about 20.00 meters for the others. The highest median of 21.61 meters is observed for the Nexus while the other test devices show a median of about 20.64 meters. The interquartile ranges vary from 8.16 meters for the Galaxy Note to 10.59 meters for the iPhone.

Based on these results, we cannot proof that a certain device has a significant impact on our estimations. We assume, that this is caused by the performed active scans, where probe requests are sent out on most devices with a similar signal power. In summary, these estimations show low accuracies based on the Fréchet Distance to ground truth. Overall, we observe a mean distance error of 19.98 meters with a standard deviation of 6.42 meters on average for all path estimations which is not suitable for indoor tracking systems. Therefore, we now investigate how the proposed probabilistic approaches improve our deterministic trajectory estimations.

### C. Probabilistic Estimation Accuracy

For comparison, we take the same 171 artificial paths and the Wi-Fi captures from the Nexus device. As distance metrics, we consider both the discrete Fréchet Distance and the average of pairwise Euclidean distances between our estimations and ground truth locations. We investigate the accuracy of all methods using the following initial parameters: the average distance between hidden states is about 6 meters leading to $|S| = 57$ state nodes within the building's corridors. The amount of particles for the SIR particle filter (PF), and our sPF, is set to $|P| = 16 \times |S|$, being 912 particles, respectively. Figure 5 depicts the results with confidence intervals base on a confidence level of 0.95. It is observed, that PF achieves the lowest accuracy on average for both distance metrics in

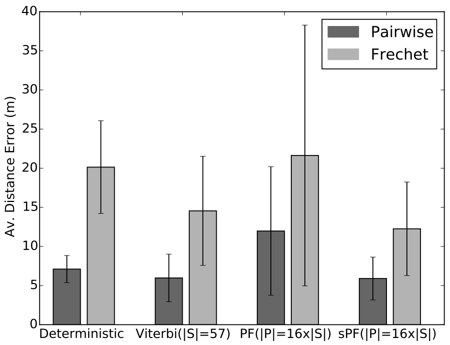
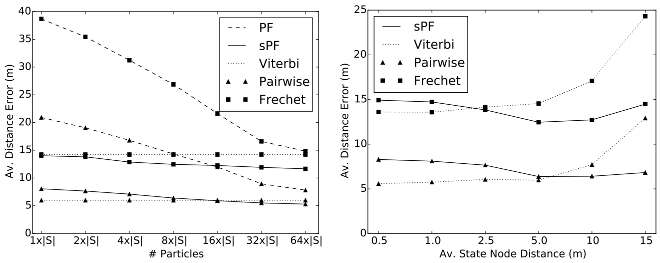


Fig. 5. Comparison of estimation errors between all methods.



(a) Varying the amount of particles (b) Varying the state node density

Fig. 6. Comparison of estimation errors of probabilistic methods.

comparison to the other approaches which have a restricted solution space. This confirms our assumption, that room-level accuracy is not achievable using passive Wi-Fi observations. Hence, it is more suitable to focus on a restricted state graph covering all possible user paths, rather than considering the complete solution space for position fixes. The results also indicate, that probabilistic methods based on an underling state graph achieve higher accuracies than the deterministic approach. We observe an improvement in terms of the Fréchet Distance of 5.59 meters in case of Viterbi, and 7.93 meters in case of sPF. Thus, the best estimation accuracy of 12.25 meters with a standard deviation of 5.97 meters is provided by the sPF followed by Viterbi achieving a mean Fréchet Distance error of 14.55 meters and a standard deviation of 6.97 meters. Considering the pairwise distance, we observe an improvement of 1.19 meters for sPF and 1.11 meters for Viterbi.

Assuming, that the amount of particles $|P|$ and the density of deployed state nodes influence the estimation accuracy, we now investigate these parameters. Hence, we firstly keep our state nodes and vary $|P|$ for both filters. Again, we consider Fréchet and the average pairwise distance and double $|P|$ from $1 \times |S|$ to $64 \times |S|$. Note, that with $|P| = 64 \times |S|$ and $|S| = 57$ the scalability is worse than in case of Viterbi.

As shown in Figure 6a, the accuracy of both filters (PF and sPF) increases with more particles, which was expected. More interesting, the Fréchet Distance error is always lower than Viterbi in case of our sPF, and consistently higher in case of PF. Furthermore, both distance errors (Frechet and pairwise) decrease more slowly in case of sPF when augmenting the

TABLE I
OVERVIEW OF ALL TRAJECTORY ESTIMATION RESULTS IN BEST CASE.

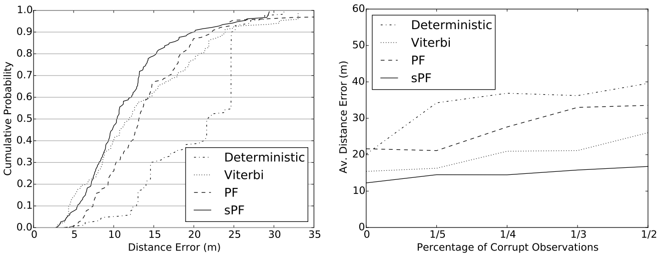
| **Used Method** | **Det.** | **Viterbi** | **PF** | **sPF** |
|---|---|---|---|---|
| **Av. dist. (Fréchet)** | 19.98 | 13.62 | 14.84 | 11.65 |
| **Std. dev. (Fréchet)** | 6.42 | 7.78 | 9.86 | 5.98 |
| **Av. dist. (Pairwise)** | 7.10 | 5.58 | 7.80 | 5.29 |
| **Std. dev. (Pairwise)** | 1.74 | 2.71 | 5.52 | 2.33 |
| **Parameters** ($\|S\|$,$\|P\|$) | n/a , n/a | 368, n/a | n/a , 3648 | 72, 3648 |

particle amount. These observations show, that the accuracy of sPF is less sensitive to $|P|$ than in case of PF. Even with the smallest possible amount of particles our sPF achieves adequate estimation results in comparison to the state-of-the art approaches. In contrast, PF requires a much higher amount (64 times) of particles to achieve a similar accuracy than Viterbi, and thus, sPF shows a higher scalability for the same accuracy level than PF, due to $O(|P|)$.

We now investigate the impact of the state node density for both Viterbi and sPF. Hence, we vary the average distance between consecutive state nodes while keeping the amount of particles constant at $|P| = 456$, which conforms to $8 \times |S|$ for the previous state node setting. Again, Figure 6b depicts the results considering the average distance errors. As expected, the accuracy of Viterbi decreases with an increasing state node distance, due to a smaller solution space for estimations. In contrast, the state particle filter shows a minimum error at an average state node distance of 5 meters, while the higher errors are observed for very dense and sparse state node settings. Thus, for smaller node distances, the particles' movement behaves more like in open space models, and for longer node distances, particles move in a very restricted way which may not represent real user traces. Hence, it is important to find an adequate state node placement with an optimal density. For this purpose, we use our heuristic, presented in Section III-C1 returning an average node distance of 6 meters which comes close to the best parameter observed here.

As depicted in Figure 6b, Viterbi performs more accurate than the sPF when the average state node distance is smaller that 2.0 meters which obviously requires a higher amount of deployed nodes (386 in our case). Take into account, the more state nodes are deployed, the higher the time complexity of the algorithm, and the lower its scalability. In contrast, the complexity of sPF is not affected by a higher amount of sate nodes, due to pre-calculation of distances and angles. Hence, if more state nodes are deployed, more particles can be used without loosing performance benefits in comparison to Vitebri. Therefore, it is worth to take a final look at the best estimation results for all of our approaches, depicted in Table I. The last row shows the chosen parameter for the amount of state nodes $|S|$, and the number of particles $|P|$, where n/a denotes that the parameter has no influence on the corresponding method.

The observed Fréchet Distance errors indicate, that all probabilistic methods help to increase the estimation accuracy compared to the deterministic (Det.) approach. Overall, our sPF achieves the highest accuracy with the lowest standard deviation in best case, and hence, it seems more feasible for Wi-Fi based indoor tracking. Based on the pairwise distance, we also observe higher accuracies for Viterbi and sPF, while the standard deviation for deterministic estimations is lower.



(a) Precision

(b) Robustness

Fig. 7. Comparison of all methods in terms of precision and robustness.

## D. Further Performance Metrics

Beside the accuracy, as one of the most important predicate for positioning systems, we now investigate further well-adopted performance metrics, such as precision and robustness. Generally, the precision is a measure of how consistently the system works. It is often defined as the standard deviation of the estimation error. However, Liu et al. [22] use the cumulative distribution function (cdf) of the distance error to measure a system's precision. Hence, we determine the cdf of our observed Fréchet distance errors for all methods base on the best parameters (cf. Table I). Figure 7a reveals the results. Obviously, all probabilistic approaches show a considerably higher precision than our deterministic trajectory estimations. As the cdf is described by percentiles, we state an estimation precision of 90% within 20.12, 22.29, 23.91, and 24.69 meters for sPF, PF, Viterbi, and the deterministic approach, respectively. In summary, our sPF shows the highest precision estimation on the conducted experiments.

Next, we investigate the robustness as a measure of how well the system reacts on corrupt, or missing signals. For passive Wi-Fi tracking, this metric is important, due to corrupt RSS measures and the arbitrary nature of probe request bursts, leading to frequent no-observations. For this purpose, we consider the observations of the 50 longest generated paths, and estimate the accuracy to ground truth for all proposed methods. We repeat this step while increasing the amount of corrupt measurements setting a certain percentage of observations to -100 dBm, which is equal to "not-observed". Figure 7b depicts the mean Fréchet Distance errors of this experiment where up to the half of our observations is being corrupted.

It can be seen, that the accuracy of all methods decreases with increasing corrupt observations. However, for evaluation of the system's robustness, the degree of the accuracy decrease has to be considered. It applies: the smaller the difference of estimation errors for increased corrupt observations, the higher the robustness. Hence, we observe that the deterministic method obtains a poor robustness, especially when a small percentage of observations, e.g. 1/5, is not valid which is very common in case of passive Wi-Fi tracking. In contrast, the

TABLE II
SUMMARY OF INVESTIGATED METHODS AND PERFORMANCE METRICS.

| **Used Method** | **Det.** | **Viterbi** | **PF** | **sPF** |
|---|---|---|---|---|
| **Mean Accuracy** | 19.98 | 13.62 | 14.84 | 11.65 |
| **Precision (90%)** | 24.69 | 23.91 | 22.29 | 20.12 |
| **Complexity** | Low | High | Moderate | Moderate |
| **Scalability** | Very Good | Low | Moderate | Good |
| **Robustness** | Poor | Moderate | Moderate | Good |

proposed state particle filter obtains the most constant results with a small difference of 4.92 meters between the lowest and the highest observed estimation error. This represents a good robustness. In comparison, we observe 10.62 meters of difference for Viterbi, and 11.93 meters for PF, which can be seen as a moderate robustness.

In summary, Table II depicts the results for all of our investigations according to Liu et al. Note, we have not evaluated the cost metric, because Wi-Fi tracking do not require any hardware modifications, and thus, it is of low-cost when performed on existing infrastructures. Based on these results, we conclude that our sPF performs best in terms of accuracy, precision, and robustness. Furthermore, a much smaller amount of particles is needed than in case of a common SIR particle filter, rendering the sPF scalable for large buildings. In comparison, a deterministic tracking method is indeed of lower complexity, and thus, more scalable. However, it has not achieved adequate results in terms of accuracy, precision and robustness within our experiments, and hence, we claim that probabilistic methods are generally required to make Wi-Fi based tracking systems more feasible.

## V. Conclusion

In this paper, we have addressed the problem of making Wi-Fi based indoor tracking systems feasible. Based on arbitrary probe request bursts from mobile devices we have performed position fixes without the user's awareness. It has been shown, that such deterministic location estimates are highly inaccurate, and it has been demonstrated how probabilistic methods can help to improve the estimation accuracy, precision, and robustness. We have investigated two common approaches, i.e., Viterbi's algorithm, and SIR particle filter, and we have presented a novel state particle filter which has achieved the best results on our experiments using real Wi-Fi captures. Overall, it has proven useful to consider a restricted movement model based on the building's corridors, rather than taking all possible user locations into account. In summary, our results indicate that passive Wi-Fi tracking becomes more accurate, scalable, and thus, feasible within indoor scenarios using the state particle filter presented in this work.

The conducted experiments have to be repeated at complexer scenarios and other settings for future work, in order to confirm our conclusions. Beside this, we have already started to evaluate the impact of the device usage where we investigate typical usage patterns for mobile devices in real world settings. First tests at our building have achieved good results in terms of accuracy when using Viterbi and our state particle filter. However, further experiments are required and corresponding results will be given in detail in future work.

## References

[1] L. Schauer, M. Werner, and P. Marcus, "Estimating crowd densities and pedestrian flows using wi-fi and bluetooth," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2014, pp. 171–177.

[2] A. Musa and J. Eriksson, "Tracking unmodified smartphones using wi-fi monitors," in *Proceedings of the 10th ACM conference on embedded network sensor systems*, 2012, pp. 281–294.

[3] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. Ieee, 2000, pp. 775–784.

[4] P. Bahl, V. N. Padmanabhan, and A. Balachandran, "Enhancements to the radar user location and tracking system," technical report, Microsoft Research, Tech. Rep., 2000.

[5] J. Krumm and E. Horvitz, "Locadio: Inferring motion and location from wi-fi signal strengths." in *Mobiquitous*, 2004, pp. 4–13.

[6] J. Trogh, D. Plets, L. Martens, and W. Joseph, "Advanced real-time indoor tracking based on the viterbi algorithm and semantic data," *International Journal of Distributed Sensor Networks*, vol. 2015, 2015.

[7] R. Lim, "Tracking smartphones using low-power sensor nodes," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2013, p. 52.

[8] A. J. Ruiz-Ruiz, H. Blunck, T. S. Prentow, A. Stisen, and M. B. Kjærgaard, "Analysis methods for extracting knowledge from large-scale wifi monitoring to inform building facility planning," in *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*. IEEE, 2014, pp. 130–138.

[9] H. Leppäkoski, J. Collin, and J. Takala, "Pedestrian navigation based on inertial sensors, indoor map, and wlan signals," *Journal of Signal Processing Systems*, vol. 71, no. 3, pp. 287–296, 2013.

[10] H. Nurminen, A. Ristimaki, S. Ali-Loytty, and R. Piché, "Particle filter and smoother for indoor localization," in *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*. IEEE, 2013.

[11] S. Bartoletti, A. Conti, A. Giorgetti, and M. Z. Win, "Sensor radar networks for indoor tracking," *Wireless Communications Letters, IEEE*, vol. 3, no. 2, pp. 157–160, 2014.

[12] M. Khan, N. Salman, A. Ali, A. Khan, and A. Kemp, "A comparative study of target tracking with kalman filter, extended kalman filter and particle filter using received signal strength measurements," in *Emerging Technologies (ICET), 2015 International Conference on*. IEEE, 2015.

[13] Q. Chen, B. Tan, K. Woodbridge, and K. Chetty, "Indoor target tracking using high doppler resolution passive wi-fi radar," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5565–5569.

[14] S. Lee, M. Kim, S. Kang, K. Lee, and I. Jung, "Smart scanning for mobile devices in wlans," in *Communications (ICC), IEEE International Conference on*, 2012, pp. 4960–4964.

[15] H. Nurminen, J. Talvitie, S. Ali-Loytty, P. Muller, E. Lohan, R. Piché, and M. Renfors, "Statistical path loss parameter estimation and positioning using rss measurements in indoor wireless networks," in *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*. IEEE, 2012, pp. 1–9.

[16] M. Kessel and M. Werner, "Smartpos: Accurate and precise indoor positioning on mobile phones," in *Mobile Services, Resources, and Users, MOBILITY, 1st International Conference on*, 2011, pp. 158–163.

[17] G. D. Forney Jr, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[18] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. ACM, 2005, pp. 205–218.

[19] B. R. S. Arulampalam, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House, 2004.

[20] M. Klepal, S. Beauregard *et al.*, "A novel backtracking particle filter for pattern matching indoor localization," in *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*. ACM, 2008, pp. 79–84.

[21] P. Marcus, "Erfassung und behandlung von positionsfehlern in standortbasierter autorisierung," Ph.D. dissertation, lmu, 2015.

[22] H. Liu, H. Darabi, P. Banerjee, and J. Liu, “Survey of wireless indoor positioning techniques and systems,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 6, pp. 1067–1080, 2007.