On Monte Carlo Smoothing in Multi Sensor Indoor Localisation

Toni Fetzer, Frank Ebner, and Frank Deinzer
Faculty of Computer Science and Business Information Systems
University of Applied Sciences Würzburg-Schweinfurt
Würzburg, Germany
{toni.fetzer, frank.ebner, frank.deinzer}@fhws.de

Lukas Köping, Marcin Grzegorzek
Pattern Recognition Group
University of Siegen
Siegen, Germany
{lukas.koeping, marcin.grzegorzek}@uni-siegen.de

***Abstract*—Indoor localisation continuous to be a topic of growing importance. Despite the advances made, several profound problems are still present. For example, estimating an accurate position from a multimodal distribution or recovering from the influence of faulty measurements. Within this work, we solve such problems with help of Monte Carlo smoothing methods, namely forward-backward smoother and backward simulation. In contrast to normal filtering procedures like particle filtering, smoothing methods are able to incorporate future measurements instead of just using current and past data. This enables many possibilities for further improving the position estimation. Both smoothing techniques are deployed as fixed-lag and fixed-interval smoother and a novel approach for incorporating them easily within a conventional localisation system is presented. All this is evaluated on four floors within our faculty building. The results show that smoothing methods offer a great tool for improving the overall localisation. Especially fixed-lag smoothing provides a great runtime support by reducing timely errors and improving the overall estimation with affordable costs.**

I. Introduction

Determining a position indoors is a challenging task. Besides the complex architecture of many buildings, high accuracy needs to be achieved, especially for buildings with many small separated areas like shopping malls or office blocks. In recent years, many different systems were presented to meet those requirements. Especially Wi-Fi positioning and pedestrian dead reckoning (PDR) are very popular solutions. Approaches based on PDR try to estimate the current position given the previous position and thus require an initial state. However, this allows for cumulative errors and leads to an erroneous position estimation within a very short period. By incorporating the absolute position information of Wi-Fi this drift can be corrected. Additional improvements can be achieved by using environmental information about walls and obstacles provided by a floor map.

In most cases, probabilistic methods are used to incorporate those highly different sensor types. Here, a probability distribution describes the pedestrian's possible whereabouts and therefore the uncertainty of the system. Drawing from a probability distribution and finding an analytical solution for densities is in most cases a difficult task, especially in case of time sequential, non-linear and non-Gaussian models. Due to the high complexity of the human movement, we consider indoor localisation as such.

Bayesian filters solve such problems by updating the state estimation recursively with every new incoming measurement. A broad class to obtain numerical results for this approach are Monte Carlo (MC) methods, where a set of random samples is used to approximate the underlying probability distribution. By applying the time sequential hidden Markov process of Bayes filtering, one of the most important MC techniques results: particle filtering. Here, a set of weighted random samples is used to solve the state estimation process.

Based on this general methodology, many different approaches for estimating a position in indoor environments have been developed. All these approaches differ mainly in how the dynamics are modelled in the transition step and how a specific sensor measurement can be used for evaluation. For example, recent approaches are using a graph-based structure to consider environmental restrictions (walking through walls) and the characteristics of human movement (walking speed) within the transition model [1]–[3]. The evaluation model is mostly separated into any number of sensor models, each representing the probability for a noisy measurement in regard to the current position. For example, a barometer can be used to determine the probability of being on a certain floor [4].

Despite the many advances made in the last years, nearly all systems suffer from more or less the same problems. Like mentioned before, PDR suffers from an accumulating bias, the signal of Wi-Fi gets attenuated by walls and the barometric pressure is highly affected by weather patterns and humidity [4]. That is the reason for the use of statistical methods in the first place. Nevertheless, there are even more profound problems regarding the whole position estimation procedure.

Current transition models, which aim to approximate the movement, are still very restrictive and unable to handle unforeseen events. Faulty sensor measurements, like a falsely detected turn, can cause the estimation to lose track. For example by recognising a turn too soon and walking into a room instead of another big hallway. Due to this, the filter needs some time to recover, which again takes a while because of the restrictive model (e.g. no walking through walls and only realistic walking speed). This temporal delay worsens the estimate immensely. A solution to recover from such filter divergences faster, involves methods to re-initialize the filtering procedure [2]. However, even this can not completely prevent

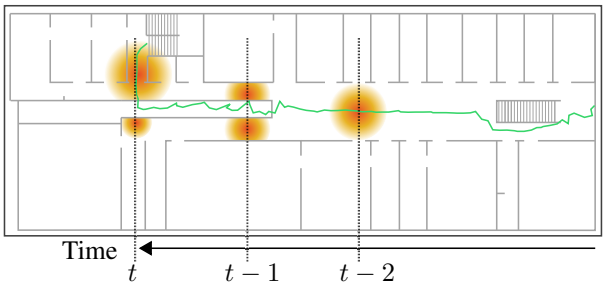

Fig. 1. An example of the occurrence of a multimodal distribution. At time $t-1$ the floor is separated by a wall and the distribution (coloured circle), splits apart. The most likely position (green line) is estimated somewhere in-between. After a right turn, the distribution slowly starts to recover its unimodality.

delays. Another reason for possible time delays are slow sensor updates. For example, most mobile devices restrict the Wi-Fi module to update only every few seconds, to save on battery. Further critical problems arise from multimodal distributions. Those are caused by multiple possible position estimates. Fig. 1 illustrates an example where a floor gets separated by a wall. Due to inaccurate measurements and a PDR approach for evaluating the movement, the distribution splits apart. Therefore, the weighted average position is somewhere in-between. Only after the pedestrian turns right, the distribution is again unimodal, since moving through walls is impossible. As one can imagine, this can lead to serious problems in big indoor environments. Such a situation can be improved by incorporating future measurements (e.g. the right turn) to the filtering procedure [5]. However, standard filtering methods are not able to use any future information and the possibilities to make a distant forecast are also limited [6]–[8].

One very promising way to deal with these problems is smoothing. Smoothing methods are able to make use of future measurements for computing their estimation. By running backwards in time, they are also able to remove multimodalities and improve the overall localisation result. Since the problem of navigation, especially the representation of complex movement patterns, results in a non-linear and non-Gaussian state space, this work focuses mainly on smoothing techniques based on the broad class of MC methods. Namely, forward-backward smoothing [9] and backward simulation [10].

Within this work, we investigate the benefits and drawbacks of those techniques using a conventional localisation system [5]. We provide both, fixed-lag and fixed-interval smoothing as well as a novel approach for incorporating them easily within the localisation procedure. Additionally, we enrich the state transition model with an activity recognition to distinguish between walking, standing and walking stairs. The main goal is to solve above mentioned problems and to investigate new possibilities for even more advanced systems. All of our contributions are supported by an extensive experimental evaluation.

II. Related Work

Sequential MC filters, like the aforementioned particle filter, use all observations $\boldsymbol{o}_{1:t}$ until the current time t for computing an estimation of the state $\boldsymbol{q}_t$. In a Bayesian setting, this can be formalized as the computation of the posterior distribution $p(\boldsymbol{q}_t \mid \boldsymbol{o}_{1:t})$ using a sample of N independent random variables, $\boldsymbol{X}_t^i \sim p(\boldsymbol{q}_t \mid \boldsymbol{o}_{1:t})$ for $i = 1, ..., N$ for approximation. Due to importance sampling, a weight W_t^i is assigned to each sample $\boldsymbol{X}_t^i$. In the context of particle filtering $\{W_{1:t}^i, \boldsymbol{X}_{1:t}^i\}_{i=1}^N$ is a weighted set of samples, also called particles. Therefore a particle is a representation of one possible system state $\boldsymbol{q}$. By considering a situation given all observations $\boldsymbol{o}_{1:T}$ until a time step T, where $t \ll T$, standard filtering methods are not able to make use of this additional data for computing $p(\boldsymbol{q}_t \mid \boldsymbol{o}_{1:T})$. This problem can be solved with a smoothing algorithm.

Within this work we utilise two types of smoothing: fixed-lag and fixed-interval smoothing. In fixed-lag smoothing, one tries to estimate the current state, given measurements up to a time $t+\tau$, where τ is a predefined lag. This makes the fixed-lag smoother able to run online. On the other hand, fixed-interval smoothing requires all observations until time T and therefore only runs offline, after the filtering procedure is finished [8].

The origin of MC smoothing can be traced back to Genshiro Kitagawa. In his work [11] he presented the simplest form of smoothing as an extension to the particle filter. This algorithm is often called the filter-smoother since it runs online and a smoothing is provided while filtering. This approach uses the particle filter steps to update weighted paths $\{(W_t^i, \boldsymbol{X}_{1:t}^i)\}_{i=1}^N$, producing an accurate approximation of the filtering posterior $p(\boldsymbol{q}_t \mid \boldsymbol{o}_{1:t})$ with a computational complexity of only $\mathcal{O}(N)$. However, it gives a poor representation of previous states due to a monotonic decrease of distinct particles caused by resampling of each weighted path [7]. Based on this, more advanced methods like the forward-backward smoother [9] and backward simulation [10] were developed. Both methods are running backwards in time to reweight a set of particles recursively by using future observations. Algorithmic details will be shown in section IV.

In recent years, smoothing got attention mainly in other areas than indoor localisation. The early work of [12] demonstrates the possibilities of smoothing for visual tracking. They used a combination of the CONDENSATION particle filter with a forward-backward smoother. Based on this pioneering approach, many different solutions for visual and multi-target tracking have been developed [13]. For example, in [14] a particle smoother is used to reduce multimodalities in a blood flow simulation for human vessels. The authors of [15] use a smoother to overcoming the problem of particle impoverishment while predicting the Remaining Useful Life (RUL) of equipment (e.g. a Lithium-ion battery).

Nevertheless, there are some promising approaches for indoor localisation systems as well. For example [2] deployed a fixed-interval forward-backward smoother to improve the position estimation for non-real-time applications. They combined Wi-Fi, step and turn detection, a simple line-of-sight

model for floor plan restrictions and the barometric change within a particle filter. The state transition samples a new state based on the heading change, altitude change and a fixed step length. The experiments of [2] clearly emphasize the benefits of smoothing techniques. The estimation error could be decreased significantly. However, a fixed-lag smoother was discussed only in theory.

In the work of [16] both fixed-interval and fixed-lag smoothing were presented. They implemented Wi-Fi, binary infrared motion sensors, binary foot-switches and a potential field for floor plan restrictions. Those sensors were incorporated using a sigma-point Kalman filter in combination with a forward-backward smoother. It was also proven by [16], that the fixed-lag smoother is slightly less accurate than the fixed-interval smoother, as one would expect from the theoretical foundation. Unfortunately, even a sigma-point Kalman filters is after all just a linearisation and therefore not as flexible and suited for the complex problem of indoor localisation as a non-linear estimator like a particle filter. Additionally, the Wi-Fi RSSI model requires known calibration points and is deployed using a remarkable number of access points for very small spaces. In our opinion this is not practical and does not suite real-world conditions. Since humans with a specific destination in mind do not tend to change their directions randomly, we would further recommend adding a PDR-based transition to draw samples in a more directed manner instead of scattering them randomly in every direction.

The here presented approach will use two different smoothing algorithm, both implemented as fixed-interval and fixed-lag versions. Further, our localisation system presented in [5] enables us to walk stairs and thus going into the third dimension. Therefore, a regularly tessellated graph is utilised to avoid walls, detecting doors and recognizing stairs. Within this work, this is additionally supported by a simple classification that detects the activities unknown, standing, walking and walking stairs.

III. Filtering

As mentioned before, most smoothing methods require a preceding filtering. Similar to our previous works, we consider indoor localisation as a time-sequential, non-linear and non-Gaussian state estimation problem. Therefore, a Bayes filter that satisfies the Markov property is used to calculate the posterior, which is given by

$$p(\boldsymbol{q}_t \mid \boldsymbol{o}_{1:t}) \propto \underbrace{p(\boldsymbol{o}_t \mid \boldsymbol{q}_t)}_{\text{evaluation}} \int \underbrace{p(\boldsymbol{q}_t \mid \boldsymbol{q}_{t-1}, \boldsymbol{o}_{t-1})}_{\text{transition}} \underbrace{p(\boldsymbol{q}_{t-1} \mid \boldsymbol{o}_{1:t-1}) d\boldsymbol{q}_{t-1}}_{\text{recursion}} \ . \tag{1}$$

Here, the previous observation $\boldsymbol{o}_{t-1}$ is included into the state transition [17]. For approximating eq. (1) by means of MC methods, the transition is used as proposal distribution, also known as CONDENSATION algorithm [12]. This algorithm also performs a resampling step to handle the phenomenon of weight degeneracy.

In context of indoor localisation, the hidden state $\boldsymbol{q}$ is defined as follows:

$$\boldsymbol{q} = (x, y, z, \theta, \hat{\rho}_{\text{rel}}), \ \ x, y, z, \theta, \hat{\rho}_{\text{rel}} \in \mathbb{R} \ , \tag{2}$$

where x, y, z represent the position in 3D space, θ the user's heading and $\hat{\rho}_{\text{rel}}$ the relative atmospheric pressure prediction in hectopascal (hPa). Further, the observation is given by

$$\boldsymbol{o} = (\boldsymbol{s}_{\text{wifi}}, \boldsymbol{s}_{\text{ib}}, \Delta\theta, n_{\text{steps}}, \rho_{\text{rel}}, \Omega) \ , \tag{3}$$

covering all relevant sensor measurements. Here, $\boldsymbol{s}_{\text{wifi}}$ and $\boldsymbol{s}_{\text{ib}}$ contain the measurements of all nearby access-points (AP) and iBeacons, respectively. $\Delta\theta$ and n_{steps} describe the relative angular change and the number of steps detected for the pedestrian. ρ_{rel} is the relative barometric pressure with respect to a fixed reference. Finally, Ω contains the activity, currently estimated for the pedestrian, which is one of: unknown, standing, walking, walking stairs up or walking stairs down.

The probability density of the state evaluation is given by

$$p(\boldsymbol{o}_t \mid \boldsymbol{q}_t) = p(\boldsymbol{o}_t \mid \boldsymbol{q}_t)_{\text{baro}} \, p(\boldsymbol{o}_t \mid \boldsymbol{q}_t)_{\text{ib}} \, p(\boldsymbol{o}_t \mid \boldsymbol{q}_t)_{\text{wifi}} \tag{4}$$

and therefore similar to [5]. Here, we assume a statistical independence of all sensors and every single component refers to a probabilistic sensor model. The barometer information is evaluated using $p(\boldsymbol{o}_t \mid \boldsymbol{q}_t)_{\text{baro}}$, whereby absolute position information is given by $p(\boldsymbol{o}_t \mid \boldsymbol{q}_t)_{\text{ib}}$ for iBeacons and by $p(\boldsymbol{o}_t \mid \boldsymbol{q}_t)_{\text{wifi}}$ for Wi-Fi.

A. Evaluation

The probability of currently residing on a floor is evaluated using the smartphone's barometer. Environmental influences are circumvented by using relative pressure values instead of absolute ones. To reduce the impact of noisy sensors, we calculate the average $\overline{\rho_{\text{rel}}}$ of several sensor readings, carried out while the pedestrian chooses his destination. This average serves as relative base for all future measurements. Likewise, we estimate the sensor's uncertainty σ_{baro} for later use within the evaluation step.

In order to evaluate the relative pressure readings, we need a prediction to compare them with. Therefore, each transition from $\boldsymbol{q}_{t-1}$ to $\boldsymbol{q}_t$ estimates the state's relative pressure prediction $\hat{\rho}_{\text{rel}}$ by tracking every height-change (z-axis):

$$q_t^{\hat{\rho}_{\text{rel}}} = q_{t-1}^{\hat{\rho}_{\text{rel}}} + \Delta z \cdot b, \ \ \Delta z = q_{t-1}^{z} - q_t^{z}, \ \ b \in \mathbb{R} \ . \tag{5}$$

In eq. (5), b denotes the common pressure change in $\frac{\text{hPa}}{\text{m}}$. The evaluation step for time t compares every predicted relative pressure $q_t^{\hat{\rho}_{\text{rel}}}$ with the observed one $o_t^{\rho_{\text{rel}}}$ using a normal distribution with the previously estimated σ_{baro}:

$$p(\boldsymbol{o}_t \mid \boldsymbol{q}_t)_{\text{baro}} = \mathcal{N}(o_t^{\rho_{\text{rel}}} \mid q_t^{\hat{\rho}_{\text{rel}}}, \sigma_{\text{baro}}^2) \ . \tag{6}$$

The smartphone's Wi-Fi and iBeacon component provides an absolute location estimation by measuring the signal-strengths of nearby transmitters. The positions of detected access-points (AP) and iBeacons are known beforehand. Using the wall-attenuation-factor signal strength prediction model [1], we are able to compare each measurement with a corresponding estimation. To infer this estimation, the prediction

model uses the 3D distance d and the number of floors Δf between transmitter and the state-in-question $\boldsymbol{q}$:

$$P_r(d, \Delta f) = P_0 - 10\gamma \log_{10} \frac{d}{d_0} + \Delta f \beta \ , \tag{7}$$

In eq. (7), there are three more parameters per AP. The signal-strength P_0 measurable at a distance d_0 (usually 1 m), a path-loss exponent γ describing the transmitter's environment and the attenuation per floor β. To reduce the system's setup time, we use the same three values for all access-points at the cost of accuracy. All parameters are chosen empirically. Further details on how to determine this parameters exactly, can be found in [18].

The same holds for the iBeacon component, except P_0, which is broadcasted by each beacon. However, as iBeacons cover only a small area, γ is usually much smaller compared to the one needed for Wi-Fi.

As transmitters are assumed to be statistically independent, the overall probability to measure their predictions at a given location is:

$$p(\boldsymbol{o}_t \mid \boldsymbol{q}_t)_{\text{wifi}} = \prod_{i=1}^{n} \mathcal{N}(s^i_{\text{wifi}} \mid P_r(d_i, \Delta f_i), \sigma^2_{\text{wifi}}) \ . \tag{8}$$

B. Transition

The transition-distribution $p(\boldsymbol{q}_t \mid \boldsymbol{q}_{t-1})$ is sampled via random walks on a graph $G = (V, E)$, which is generated from the buildings floorplan [5]. Each walk starts at $\boldsymbol{q}_{t-1}$ and uses adjacent edges $e_{i,j}$ connecting two vertices $v_i, v_j \in V$ until a certain distance d_{step} is reached. Hereby, the position of any $\boldsymbol{q}$ is represented by the position $\mathbf{pos}(v_i)$ of its corresponding vertex v_i. This approach draws only valid movements, as ambient conditions (walls, doors, stairs, etc.) are considered.

While sampling, to-be-walked edges are not chosen uniformly, but depending on a probability $p(e_{i,j})$. The latter depends on several constraints and recent sensor-readings from the smartphone. Using those readings directly within the transition step provides a more robust posterior distribution. Adding them to the evaluation instead, would lead to sample impoverishment due to the used Monte Carlo methods [19].

Steps and turns are detected using the smartphone's IMU, implemented as described in [1]. The number of steps detected since the last transition is used to estimate the to-be-walked distance d_{step} by assuming a fixed step-size with some deviation:

$$d_{\text{step}} = o^{n_{\text{steps}}}_{t-1} \cdot s_{\text{step}} + \mathcal{N}(0, \sigma^2_{d_{\text{step}}}) \ . \tag{9}$$

Turn-Detection supplies the magnitude of the detected heading change by integrating over the gyroscope's change since the last transition. Together with some deviation and the state's previous heading, the magnitude is used to estimate the current state's heading:

$$\theta_{\text{walk}} = q^\theta_t = q^\theta_{t-1} + o^{\Delta\theta}_{t-1} + \mathcal{N}(0, \sigma^2_{\theta_{\text{walk}}}) \tag{10}$$

During the random walk, edges should satisfy the current heading and are thus drawn according to their resemblance:

$$p(e_{i,j})_{\text{head}} = p(e_{i,j} \mid \theta_{\text{walk}}) = \mathcal{N}(\angle e_{i,j} \mid \theta_{\text{walk}}, \sigma^2_{\text{head}}) \ . \tag{11}$$

While the distribution eq. (11) does not integrate to 1.0 due to circularity of angular data, in our case, the normal distribution can be assumed as sufficient for small enough σ^2.

Additionally we perform a simple activity detection for the pedestrian, able to distinguish between several actions $\Omega \in \{\text{unknown}, \text{standing}, \text{walking}, \text{stairs\_up}, \text{stairs\_down}\}$. For this, the sensor signals are split in sliding windows. Each window has a length of one second and overlaps 500 ms with its prior window. We use a naive Bayes classifier with two features. The first one is the variance of the accelerometer's magnitude within a window. The second feature is the difference between the last and first barometer measurement of the particular window. Based on these features the classifier assigns an activity to each of the sliding windows. Similarly to the above, this knowledge is then evaluated when walking the grid: Edges $e_{i,j}$ matching the currently detected activity are favoured using $p(e_{i,j})_{\text{act}} = 0.8$ and 0.2 otherwise. If no information of the current activitiy could be obtained, no influence is exerted on the edges.

IV. Smoothing

The main purpose of this work is to provide MC smoothing methods in context of indoor localisation. As mentioned before, those algorithms are able to compute probability distributions in the form of $p(\boldsymbol{q}_t \mid \boldsymbol{o}_{1:T})$ and are therefore able to make use of future observations between t and T, where $t \ll T$. In the following we discuss the algorithmic details of the forward-backward smoother and the backward simulation. Further, a novel approach for incorporating them into the localisation system is shown.

A. Forward-backward Smoother

The forward-backward smoother (FBS) of [20] is a well established alternative to the simple filter-smoother. The foundation of this algorithm was again laid by Kitagawa in [21]. An approximation is given by

$$p(\boldsymbol{q}_t \mid \boldsymbol{o}_{1:T}) \approx \sum_{i=1}^{N} W^i_{t|T} \delta_{\boldsymbol{X}^i_t}(\boldsymbol{q}_t) \ , \tag{12}$$

where $p(\boldsymbol{q}_t \mid \boldsymbol{o}_{1:T})$ has the same support as the filtering distribution $p(\boldsymbol{q}_t \mid \boldsymbol{o}_{1:t})$, but the weights are different. This means, that the FBS maintains the original particle locations and just reweights the particles to obtain a smoothed density. $\delta_{\boldsymbol{X}^i_t}$ denotes the Dirac delta function. The complete FBS can be seen in algorithm IV-A in pseudo-algorithmic form. At first, the algorithm obtains the filtered distribution (particles) by deploying a forward step at each time t. Then the backward step for determining the smoothing distribution is carried out. The weights are obtained through the backward recursion in line 9.

By reweighting the filter particles, the FBS improves the simple filter-smoother by removing its dependence on the inheritance (smoothed) paths [22]. However, by looking at algorithm IV-A it can easily be seen that this approach computes in $\mathcal{O}(N^2)$, where the calculation of each particle's weight is an $\mathcal{O}(N)$ operation. To reduce this computational

Algorithm 1 Forward-Backward Smoother

for $t = 1$ **to** T **do** ▷ Filtering
  Obtain the weighted trajectories $\{W_t^i, \boldsymbol{X}_t^i\}_{i=1}^N$
end for
for $i = 1$ **to** N **do** ▷ Initialization
  Set $W_{T|T}^i = W_T^i$
end for
for $t = T - 1$ **to** 1 **do** ▷ Smoothing
  for $i = 1$ **to** N **do**
    $W_{t|T}^i = W_t^i \left[\sum_{j=1}^N W_{t+1|T}^j \frac{p(\boldsymbol{X}_{t+1}^j | \boldsymbol{X}_t^i)}{\sum_{k=1}^N W_t^k \; p(\boldsymbol{X}_{t+1}^j | \boldsymbol{X}_t^k)} \right]$
  end for
end for

Algorithm 2 Backward Simulation Smoothing

for $t = 1$ **to** T **do** ▷ Filtering
  Obtain the weighted trajectories $\{W_t^i, \boldsymbol{X}_t^i\}_{i=1}^N$
end for
for $k = 1$ **to** N_{sample} **do**
  Choose $\tilde{\boldsymbol{q}}_T^k = \boldsymbol{X}_T^i$ with probability W_T^i ▷ Initialize
  for $t = T - 1$ **to** 1 **do** ▷ Smoothing
    for $j = 1$ **to** N **do**
      $W_{t|t+1}^j = W_t^j \; p(\tilde{\boldsymbol{q}}_{t+1} \mid \boldsymbol{X}_t^j)$
    end for
    Choose $\tilde{\boldsymbol{q}}_t^k = \boldsymbol{X}_t^j$ with probability $W_{t|t+1}^j$
  end for
  $\tilde{\boldsymbol{q}}_{1:T}^k = (\tilde{\boldsymbol{q}}_1^k, \tilde{\boldsymbol{q}}_2^k, ..., \tilde{\boldsymbol{q}}_T^k)$ is one approximate realisation from $p(\boldsymbol{q}_{1:T} \mid \boldsymbol{o}_{1:T})$
end for

bottleneck, [23] introduced a solution using algorithms from N-body simulation. By integrating dual tree recursions and fast multipole techniques with the FBS, a run-time cost of $\mathcal{O}(N \log N)$ can be achieved.

B. Backward Simulation

For smoothing applications with a high number of particles, it is often not necessary to use all particles for smoothing. This decision can, for example, be made due to a high sample impoverishment and/or highly accurate sensors. By choosing a good sub-set for representing the posterior distribution, it is theoretically possible to further improve the estimation.

Therefore, [10] presented the backward simulation (BS). Where a number of independent sample realisations from the entire smoothing density are used to approximate the smoothing distribution. This method can be seen in algorithm 2 in pseudo-algorithmic form. Again, a particle filter is performed at first and then the smoothing procedure gets applied. Here, $\tilde{\boldsymbol{q}}_t$ is a random sample drawn approximately from $p(\boldsymbol{q}_t \mid \tilde{\boldsymbol{q}}_{t+1}, \boldsymbol{o}_{1:T})$. For example $\tilde{\boldsymbol{q}}_t$ could be chosen by selecting particles within a cumulative frequency. Therefore $\tilde{\boldsymbol{q}}_{1:T} = (\tilde{\boldsymbol{q}}_1, \tilde{\boldsymbol{q}}_2, ..., \tilde{\boldsymbol{q}}_T)$ is one particular sample realisation from $p(\boldsymbol{q}_{1:T} \mid \boldsymbol{o}_{1:T})$. Further independent realisations are obtained by repeating the algorithm until the desired number of realisations N_{sample} is reached. The computational complexity for one particular realisation is $\mathcal{O}(N)$. However, the computations are then repeated for each realisation drawn [10].

C. Transition for Smoothing

As seen above, both algorithms are reweighting particles based on a state transition model. Unlike the transition presented in section III-B, it is not possible to just draw a set of new samples. Here, $p(\boldsymbol{q}_{t+1} \mid \boldsymbol{q}_t)$ needs to provide the probability of the *known* future state $\boldsymbol{q}_{t+1}$ under the condition of its ancestor $\boldsymbol{q}_t$. The smoothing transition model therefore calculates the probability of being in a state $\boldsymbol{q}_{t+1}$ in regard to previous states and the pedestrian's walking behaviour. This means that a state $\boldsymbol{q}_t$ is more likely if it is a proper ancestor (realistic previous position) of a future state $\boldsymbol{q}_{t+1}$. In the following a simple and inexpensive approach for receiving this information will be described.

By writing

$$p(\boldsymbol{q}_{t+1} \mid \boldsymbol{q}_t, \boldsymbol{o}_t)_{\text{step}} = \mathcal{N}(\Delta d_t \mid \mu_{\text{step}}, \sigma_{\text{step}}^2) \tag{13}$$

we receive a statement about how likely it is to cover a distance Δd_t between two states $\boldsymbol{q}_{t+1}$ and $\boldsymbol{q}_t$. In the easiest case, Δd_t is the euclidean distance between two states. Of course, based on the graph structure, one could calculate the shortest path between both and sum up the respective edge lengths. However, this requires tremendous calculation time for negligible improvements. Therefore this is not further discussed within this work. The average step length μ_{step} is based on the pedestrian's walking speed and σ_{step}^2 denotes the step length's variance. Both values are chosen depending on the activity Ω recognized at time t. For example μ_{step} gets smaller while a pedestrian is walking upstairs, than just walking straight. This requires to extend the smoothing transition by the current observation $\boldsymbol{o}_t$. Since $\boldsymbol{q}$ is hidden and the Markov property is satisfied, we are able to do so.

The heading information is incorporated using

$$p(\boldsymbol{q}_{t+1} \mid \boldsymbol{q}_t, \boldsymbol{o}_t)_{\text{turn}} = \mathcal{N}(\Delta\alpha_t \mid \Delta\theta, \sigma_{\text{turn}}^2) \;, \tag{14}$$

where $\Delta\alpha_t$ is the absolute angle between $\boldsymbol{q}_{t+1}$ and $\boldsymbol{q}_t$ in the range of $[0, \pi]$. The relative angular change $\Delta\theta$ is then used to receive a statement about how likely it is to walk in that particular direction. Again the normal distribution of eq. (14) does not integrate to 1.0. Therefore the same assumption as in eq. (11) has to be made.

To further improve the results, especially in 3D environments, the vertical (non-absolute) distance Δz between two successive states is used as follows:

$$p(\boldsymbol{q}_{t+1} \mid \boldsymbol{q}_t, \boldsymbol{o}_t)_{\text{baro}} = \mathcal{N}(\Delta z \mid \mu_z, \sigma_z^2) \;. \tag{15}$$

This assigns a low probability to false detected or misguided floor changes. Similar to eq. (13) we set μ_z and σ_z^2 based on the activity recognised at time t. Therefore, μ_z is the expected change in z-direction between two time steps. This means, if the pedestrian is walking alongside a corridor, we set $\mu_z = 0$. In contrast, μ_z is positive while walking downstairs or otherwise negative for moving upstairs. The size of μ_z and

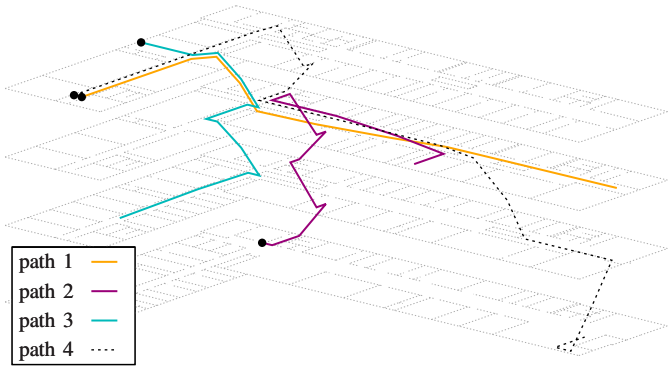

Fig. 2. The four paths that were part of the evaluation. Starting positions are marked with black circles. For a better visualisation they were slightly shifted to avoid overlapping.

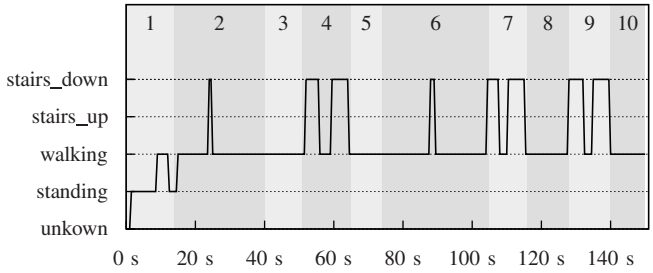

Fig. 3. The activities recognized for path 4. The misdetection in seg. 2 is cause by faulty pressure readings.

also μ_{step} could be a predefined value or set dynamically based on the measured vertical and linear acceleration.

Looking at eq. (13) to eq. (15), obvious similarities to a sensor fusion process can be seen. By assuming statistical independence between those three, the probability density of the smoothing transition is given by

$$\begin{aligned} p(\boldsymbol{q}_{t+1} \mid \boldsymbol{q}_t, \boldsymbol{o}_t) = & \; p(\boldsymbol{q}_{t+1} \mid \boldsymbol{q}_t, \boldsymbol{o}_t)_{\text{step}} \\ & p(\boldsymbol{q}_{t+1} \mid \boldsymbol{q}_t, \boldsymbol{o}_t)_{\text{turn}} \\ & p(\boldsymbol{q}_{t+1} \mid \boldsymbol{q}_t, \boldsymbol{o}_t)_{\text{baro}} \end{aligned} \quad . \tag{16}$$

It is important to notice, that all particles at each time step t of the forward filtering need to be saved. Therefore, the memory requirement increases proportional to the processing time.

V. Experiments

The experiments were carried out on all floors (0 to 3) of the faculty building. Each floor is about 77 m x 55 m in size, with a ceiling height of 3 m. To resemble real-world conditions, the evaluation took place during an in-house exhibition. Thus, many places were crowded and Wi-Fi signals attenuated. As can be seen in fig. 2 we arranged 4 distinct walks, covering different distances, critical sections and uncertain decisions leading to multimodalities. The ground truth is measured by recording a timestamp at marked spots on the walking route. When passing a marker, the pedestrian clicked a button on the smartphone application. Between two consecutive points, a constant movement speed is assumed. Thus, the ground truth might not be 100 % accurate, but fair enough for error measurements. The approximation error is then calculated by comparing the interpolated ground truth position with the current estimation. Especially in the context of smoothing, it is also very interesting to exclude the temporal delay from the error calculations and measure only the positional difference between estimated and ground truth path. This gives a statement about the extent to which the smoothed path superficially improves compared to the filtered one.

All walks start with a uniform distribution (random position and heading) as prior for $\boldsymbol{q}_0$. To allow the system to stabilize its initial state, the first few estimations are omitted from error calculations. Even though, the error during the following few seconds is expected to be much higher than the error when starting with a well known initial position and heading.

The measurements were recorded using a Motorola Nexus 6 and a Samsung Galaxy S5. As the Galaxy's Wi-Fi can not be limited to the 2.4 GHz band only, its scans take much longer than those of the Nexus: 3500 ms vs. 600 ms. Additionally, the Galaxy's barometer sensor provides fare more inaccurate and less frequent readings than the Nexus does. This results in a better localisation using the Nexus smartphone. The computation for both filtering and smoothing was done offline using the aforementioned CONDENSATION algorithm and multinomal (cumulative) resampling. For each path we deployed 10 MC runs using 2500 particles. BS uses 500 sample realisations drawn with a cumulative frequency. Unless explicitly stated, the state was estimated using the weighted arithmetic mean of the particle set.

As mentioned earlier, the position of all access-points (about 5 per floor) is known beforehand. Due to legal terms, we are not allowed to depict their positions and therefore omit this information within the figures. Additionally, we used five iBeacons for slight enhancements in some areas. The empirically chosen values for Wi-Fi were $P_{0_{\text{wifi}}} = -46\,\text{dBm}$, $\gamma_{\text{wifi}} = 2.7$, $\beta_{\text{wifi}} = 8\,\text{dB}$, and $\gamma_{\text{ib}} = 1.5$ for the iBeacons, respectively. The system was tested by omitting any time-consuming calibration processes for those values. We therefore expect the localisation process to perform generally worse compared to standard fingerprinting methods [24]. However, smoothing will often compensate for those poorly chosen system parameters.

For the filtering we used $\sigma_{\text{wifi}} = \sigma_{\text{ib}} = 8.0$ as uncertainties, both growing with each measurement's age. While the pressure change was assumed to be $0.105\,\frac{\text{hPa}}{\text{m}}$, all other barometer-parameters are determined automatically (see III-A). The step size s_{step} for the transition was configured to be 70 cm with an allowed derivation of 10 %. The heading deviation was set to 25 °. Edges departing from the pedestrian's destination are downvoted using $\kappa = 0.9$. For smoothing we set $\sigma^2_{\text{turn}} = 5\,°$ and $\sigma^2_z = 0.25\,\text{cm}$. If walking or unknown are the current activities, $\mu_{\text{step}} = 0.7\,\text{m}$, $\mu_{\text{step}} = 0.5$ and $\mu_z = 0.0\,\text{m}$ are used. Walking upstairs sets $\mu_{\text{step}} = 0.4\,\text{m}$, $\sigma^2_{\text{step}} = 0.2$ and $\mu_z = -0.3\,\text{m}$, otherwise $\mu_{\text{step}} = 0.5\,\text{m}$, $\sigma^2_{\text{step}} = 0.3$ and $\mu_z = 0.3\,\text{m}$ for walking downstairs.

By adding the activity recognition the approximation error

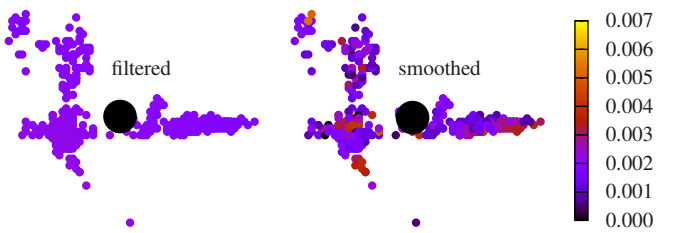

Fig. 4. Comparison between the filtered and FBS particle set. Both have identical positions and are recorded at the same time step on path 4. The black dot indicates the estimation using the weighted arithmetic mean of all particles. The different colours represent the current weight of a certain particle.

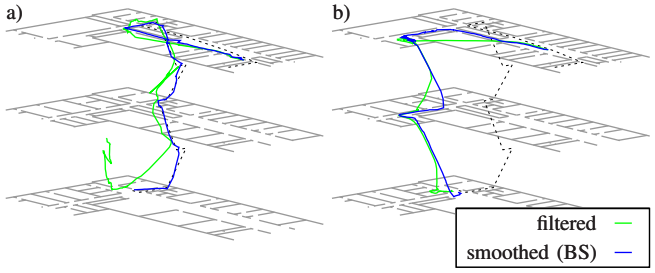

Fig. 5. a) Exemplary results for path 2 where BS (blue) and filtering (green) using 2500 particles and 500 sample realisations. b) A situation where smoothing provides a worse error in regard to the ground truth, but obviously a more realistic path.

of the filter decreases by an average of XX cm for all 4 paths. Due to this additional knowledge, the state transition samples mostly depending upon the current activity and therefore limits the possibility of false floor changes. Fig. 3 shows the recognized activities for path 4 using the Nexus 6. Despite a short misdetection in seg. 2, caused by faulty pressure readings, the recognition can be considered to be very robust and accurate.

At first, both FBS and BS are compared in context of fixed-interval smoothing. As a reminder, fixed-interval smoother are using all observations until time T and therefore run offline, after the filtering procedure is finished. Thus, we calculate only the positional error between estimation and ground truth, since timely information are negligible. In contrast to BS, the FBS is not able to improve the results using the weighted arithmetic mean for estimating the current position. Fig. 4 illustrates the filtered and smoothed particle set at a certain time step on path 4. It can be seen that the estimated position (black dot) for filtering and FBS is identical although the particle weights are highly different. To address this problem, we are extending the FBS by adding a resampling step. Here, particles with large weights are likely to be drawn multiple times. This focuses the computational resources of the FBS into regions with high probability mass and finally improves the estimation. Since smoothing operates on known states, the danger of sample impoverishment is negligible.

We deployed a multinomial resampling step to every smoothing interval of the FBS. Now, the positional average error along all 4 paths using the Nexus and the Galaxy could be improved from 2.08 m to 1.37 m. The BS performs with an average error of 2.21 m for filtering and 1.51 m for smoothing. The difference between both filtering steps is of course based upon the randomized behaviour of the respective probabilistic models. It is interesting to note, that the positional error is very similar for both used smartphones, although the approximation error varies greatly. Using the FBS, the Galaxy donates an average approximation error of 4.03 m by filtering with 7.74 m. In contrast the Nexus 6 filters at 5.11 m and results in 3.87 m for smoothing. The BS has a similar improvement rate.

Two visual examples of the smoothing outcome for path 2 are illustrated in fig. 5. It can be clearly seen, how the smoothing compensates for the faulty detected floor changes using

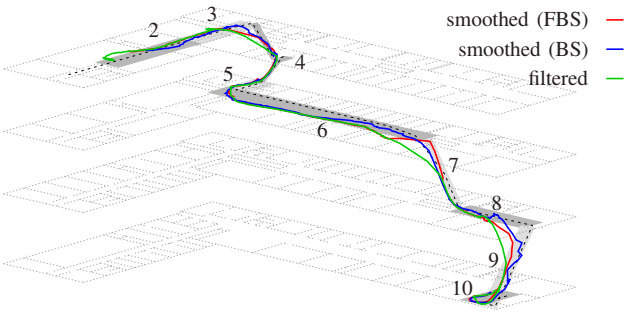

Fig. 6. Estimation results on path 4 for the filter and both smoothers using fixed-lag smoothing with $\tau = 5$. For a better visualisation, the segments are divided using an outline of alternating grey levels. The corresponding segment-error can be seen in fig. 7.

future knowledge. Additionally, the initial error is reduced extremely, approximating the pedestrian's starting position down to a few centimetres. In the context of reducing the error as far as possible, fig. 5 b) is a very interesting example. Here, the filter offers a lower approximation and positional error in regard to the ground truth. However it is obvious that smoothing causes the estimation to behave more natural instead of walking the supposed path. This phenomena could be observed for both smoothers respectively.

At next, we discuss the advantages and disadvantages of utilizing FBS and BS as fixed-lag smoother. Compared to fixed-interval smoothing, timely errors are now of higher importance due to an interest on real-time localization. Especially interesting in this context are small lags $\tau < 10$ considering filter updates near 500 ms. Fig. 6 illustrates the different estimations for path 4 using a fixed-lag $\tau = 5$. The associated approximation errors alongside the path can additionally be seen in fig. 7. Due to the small number of sample realisations for BS and the additional resampling for FBS, the errors are changing very frequently in contrast to the filter. For better distinction, the path was divided into 10 individual segments. Again it can be observed, that both smoother enable a better overall estimation especially in areas where the user is changing floors (cf. fig. 7 seg. 4, 7). Immediately after the first floor change, a long and straight walk down the hallway follows. While the Wi-Fi component pulls the pedestrian into the rooms on the right side, the actual

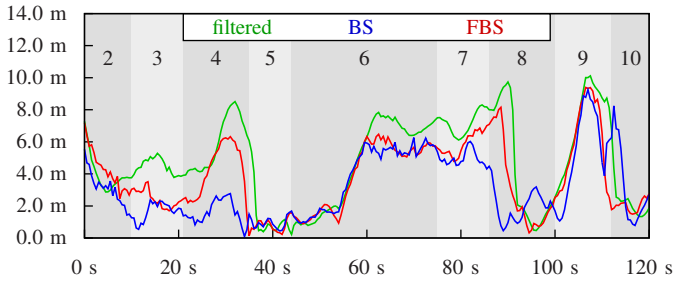

Fig. 7. Error development while walking along Path 4 using the Nexus 6. Especially in segments including floor changes, the error is reduced visibly by using smoothing methods.

walking route was located on the left side of the floor (see ground truth in fig. 6 seg. 6). Here, the BS is able to slightly improve the path, whereas the FBS follows the filtering until the upcoming staircase provides the necessary information for adjustments. By looking at fig. 6 seg. 9 it seems that both smoothing methods are highly improving the error. However, the approximation error in this area is similar to the filter and only the positional error decreases. This timely error is caused by a phenomenon we call Wi-Fi jump. Especially in seg. 8 and 9 a big crowd was gathered and highly attenuated the Wi-Fi signal. For an excessive amount of time, the absolute location estimated by the Wi-Fi component got stuck in the middle of seg. 8 and therefore delayed the estimation. The next viable measurements were then provided at the end of seg. 9. This suggests that the here presented smoothing transition is able to improve the estimated path visibly, but does not compensate for those jumps in a timely manner. Finally, the BS provides an approximation error alongside all paths of $6.48\,\mathrm{m}$ for the Galaxy and $4.47\,\mathrm{m}$ for the Nexus by filtering with $7.92\,\mathrm{m}$ and $5.50\,\mathrm{m}$ respectively. Whereas FBS improves the Galaxy's estimation from $7.73\,\mathrm{m}$ to $6.68\,\mathrm{m}$ and from $5.66\,\mathrm{m}$ to $4.80\,\mathrm{m}$ for the Nexus. As stated before, the main advantage of BS over FBS is the better computational time by just using a sub-set of particles for calculations.

Reducing the number of particles down to 500 does not necessarily worsen the estimation. In most cases smoothing compensates for this reduction and maintains the good results. Besides changing the number of particles, it is also possible the variate the lag. As one would expect, increasing the lag causes the smoothed estimation to approach the results provided by fixed-interval smoothing. It is obvious that a lag of 30 time steps has access to much more future observations and is therefore able to obtain such a result. Considering an update interval of $500\,\mathrm{ms}$, a lag of 30 would however mean that the smoother is $15\,\mathrm{s}$ behind the filter. Nevertheless, there are practical applications like accurately verifying hit checkpoints or continuously optimizing a recurring segment of the path.

VI. Conclusion

Within this work a novel approach for utilizing the forward-backward smoother and backward simulation to problems of indoor localisation was presented. Both were implemented as fixed-lag and fixed-interval smoother. It was shown that smoothing methods are able to decrease the estimation error and improving the overall localisation. Especially fixed-lag smoothing is a great tool for runtime support by reducing timely errors and improving the overall estimation with affordable costs. However, a fixed-lag smoother is not able to change the lag dynamically, as its name suggests. Therefore, a dynamic-lag smoother could be able to further improve the estimation by considering higher lags in critical areas.

Finally, the smoothing transition does not use any information provided by the underlying graph structure. This would allow to use environmental informations and to replace the current line-of-sight model with a graph-based one. By incorporating the Wi-Fi's signal strength measurements a more advanced smoothing transition should be able to compensate for faulty Wi-Fi measurements and the hereby resulting jumps between positions.

References

[1] F. Ebner, T. Fetzer, L. Köping, M. Grzegorzek, and F. Deinzer, "Multi Sensor 3D Indoor Localisation," in *Indoor Positioning and Indoor Navigation (IPIN), Int. Conf. on*, 2015, pp. 1–10.

[2] H. Nurminen, M. Koivisto, S. Ali-Loytty, and R. Piche, "Motion Model for Positioning with Graph-based Indoor Map," in *Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2014, pp. 646–655.

[3] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, and E. Steinbach, "Graph-based Data Fusion of Pedometer and WiFi Measurements for Mobile Indoor Positioning," in *Proc. of the 2014 ACM Int. Joint Conf. on Pervasive and Ubiquitous Computing - UbiComp '14 Adjunct*. New York, NY, USA: ACM Press, 2014, pp. 147–158.

[4] B. Li, B. Harvey, and T. Gallagher, "Using barometers to determine the height for indoor positioning," in *Indoor Positioning and Indoor Navigation (IPIN), Int. Conf. on*, 2013, pp. 1–7.

[5] F. Ebner, T. Fetzer, M. Grzegorzek, and F. Deinzer, "On Prior Navigation Knowledge in Multi Sensor Indoor Localisation," in *On Review*, 2016, pp. 1–8.

[6] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: The MIT Press, 2005.

[7] A. Doucet and A. M. Johansen, "A Tutorial on Particle Filtering and Smoothing: Fifteen years later," in *The Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovsky, Eds. Oxford University Press, 2011.

[8] Z. Chen, "Bayesian filtering: From kalman filters to particle filters, and beyond," *Statistics*, vol. 182, no. 1, pp. 1–69, 2003.

[9] A. Doucet, S. Godsill, and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[10] S. J. Godsill, A. Doucet, and M. West, "Monte carlo smoothing for nonlinear time series," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 156–168, 2004.

[11] G. Kitagawa, "Monte carlo filter and smoother for non-gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996.

[12] M. Isard and A. Blake, "A Smoothing Filter for Condensation," in *Computer VisionECCV'98*. Springer, 1998, pp. 767–781.

[13] P. Pérez, J. Vermaak, and A. Blake, "Data Fusion for Visual Tracking with Particles," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 495–513, 2004.

[14] E.-S. Platzer, F. Deinzer, D. Paulus, and J. Denzler, "3D Blood Flow Reconstruction from 2D Angiograms," in *Bildverarbeitung für die Medizin 2008*, ser. Informatik Aktuell, T. Tolxdorff, J. Braun, T. M. Deserno, A. Horsch, H. Handels, and H.-P. Meinzer, Eds. Springer Berlin Heidelberg, 2008, pp. 288–292.

[15] Y. Hu, P. Baraldi, F. Di Maio, and E. Zio, "A Particle filtering and Kernel Smoothing-Based Approach for New Design Component Prognostics," *Reliability Engineering and System Safety*, vol. 134, pp. 19–31, 2014.

[16] A. S. Paul and E. A. Wan, "RSSI-Based Indoor Localization and Tracking using Sigma-Point Kalman Smoothers," *IEEE Journal on Selected Topics in Signal Processing*, vol. 3, no. 5, pp. 860–873, 2009.
[17] L. Köping, M. Grzegorzek, and F. Deinzer, "Probabilistic Step and Turn Detection in Indoor Localization," in *Conf. on Data Fusion and Target Tracking 2014: Algorithms and Applications (DFTT 2014)*, Liverpool, UK, 2014, pp. 1–7.
[18] S. Y. Seidel and T. S. Rappaport, "914 MHz path loss prediction models for indoor wireless communications in multifloored buildings," *IEEE Transactions on Antennas and Propagation*, vol. 40, no. 2, pp. 207–217, 1992.
[19] M. Isard and A. Blake, "CONDENSATION - Conditional Density Propagation for Visual Tracking," *Int. Journal of Computer Vision*, vol. 29, no. 1, pp. 5–28, 1998.
[20] A. Doucet, S. Godsill, and C. Andrieu, "On Sequential Monte Carlo Sampling Methods for Bayesian Filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
[21] G. Kitagawa, "Non-gaussian state-space modeling of nonstationary time series," *Journal of the American Statistical Association*, vol. 82, no. 400, pp. 1032–1041, 1987.
[22] P. Fearnhead, D. Wyncoll, and J. Tawn, "A sequential smoothing algorithm with linear computational cost," *Biometrika*, vol. 97, no. 2, pp. 447–464, 2010.
[23] M. Klaas, M. Briers, N. De Freitas, A. Doucet, S. Maskell, and D. Lang, "Fast particle smoothing: If i had a million particles," in *Proc. of the 23rd Int. Conf. on Machine learning*. ACM, 2006, pp. 481–488.
[24] V. Honkavirta, T. Perl, S. Ali-Lytty, and R. Piche, "A comparative survey of wlan location fingerprinting methods," *Workshop on Positioning, Navigation and Communication*, vol. 6, no. 1, pp. 243–251, 2009.