Author(s)	Nurminen, Henri; Ristimäki, Anssi; Ali-Löytty, Simo; Piché, Robert
Title	Particle filter and smoother for indoor localization
Citation	Nurminen, Henri; Ristimäki, Anssi; Ali-Löytty, Simo; Piché, Robert 2013. Particle filter and smoother for indoor localization. International Conference on Indoor Positioning and Indoor Navigation, IPIN 2013, 28-31 Oct 2013, Montbéliard-Belfort, France. International Conference on Indoor Positioning and Indoor Navigation Piscataway, NJ, 137-146.
Year	2013
DOI	
Version	Post-print
URN	http://URN.fi/URN:NBN:fi:tty-201403051121
Copyright	© 2013 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Particle filter and smoother for indoor localization

Henri NURMINEN, Anssi RISTIMÄKI, Simo ALI-LÖYTTY and Robert PICHÉ
Tampere University of Technology, Tampere, Finland
Emails: {henri.nurminen, anssi.ristimaki, simo.ali-loytty, robert.piche}@tut.fi

***Abstract*—We present a real-time particle filter for 2D and 3D hybrid indoor positioning. It uses wireless local area network (WLAN) based position measurements, step and turn detection from a hand-held inertial sensor unit, floor plan restrictions, altitude change measurements from barometer and possibly other measurements such as occasional GNSS fixes. We also present a particle smoother, which uses future measurements to improve the position estimate for non-real-time applications. A light-weight fallback filter is run in the background for initialization, divergence monitoring and possibly re-initialization. In real-data tests the particle filter is more accurate and consistent than the methods that do not use floor plans. An example is shown on how smoothing helps to improve the filter estimate. Moreover, a floor change case is presented, in which the filter is capable of detecting the floor change and improving the 2D accuracy using the floor change information.**

keywords: indoor positioning; framework for hybrid positioning; particle filtering; particle smoothing; signal strength based methods

I. Introduction

Accurate real-time indoor positioning relying solely on existing architecture that has not been built for positioning purposes remains a challenging technical problem. An inexpensive wide-availability solution is wireless local area network (WLAN) positioning, which, however, suffers from poor accuracy in many cases especially if no positioning-specific modifications are done to the WLAN network. Inertial motion sensors (IMS) such as accelerometer and gyroscope are usable for pedestrian dead reckoning (PDR), which estimates the current position given the previous position. If the initial state is given, IMS can typically estimate the user's path accurately only over a short period because the biased error accumulates over time. Using time-series algorithms, WLAN positioning and PDR can be combined so that the absolute position information of WLANs is used to correct the drift of the PDR. One way to improve the accuracy further by time-series is to incorporate floor plans. In indoor positioning the floor plans include information about forbidden paths, such as paths crossing walls or other obstacles. The floor plan information results in non-linear motion models, whose estimation is very challenging in analytical methods.

One solution for the hybrid time-series positioning problem with nonlinear models is the particle filter, which is based on Bayesian statistical theory and Monte Carlo (MC) simulation. Particle filters provide a set of weighted MC samples of the state at each time instant [1, Ch. 2.5.4]. These samples are called particles. In conventional hybrid indoor positioning, the MC samples of position and possibly other position-related components are updated using the PDR information. WLANs and floor plan are then used to enhance the estimate by updating the particle weights. This approach with various modifications has been used in [2–8], among others.

We have implemented a particle filter that uses accelerometers, gyroscope, WLAN received signal strengths (RSS), floor plans and barometer. One of the novel features of our filter is divergence detection and re-initialization based on running a light-weight fallback filter in the background of the particle filter. This scheme aims to remedy some shortcomings of the approaches in [7] and [9]. Another novel feature is that, because the positioning is intended for hand-held rather than foot-mounted devices, we use the IMS only for step detection and measuring heading changes, and estimate the step length in the particle filter. We also present a method and some results of hybrid multi-floor indoor positioning using particle filter.

We also pay special attention to the use of *consistent* measurement models, that is, the dispersions of the measurement distributions are required to be realistic. We infer the WLAN position using RSS measurements and simple path loss (PL) models whose parameters as well as the access point positions are estimated off-line using learning data. We also infer the measurement uncertainty based on the WLAN environment and the uncertainties of the PL parameters. This approach enables better consistency than some previous papers that assume constant covariance for the WLAN position measurements, e.g. [2, 5, 10].

Particle smoothing for non-realtime estimation of the track of the user is also covered in this paper. The approach of [4] stores the particle histories and uses the weights of the current particles also for historical particles. A problem with this approach is that approximations of the smoothed distributions tend to be degenerate. Our smoothing method is based on forward-backward recursions [11, p. 204], and this algorithm does not suffer as much from the degeneracy problem.

We test our filter and smoother in a building at Tampere University of Technology, Finland using the existing WLAN architecture and a hand-held inertial sensor unit including barometer. No compass is used. To evaluate the importance of map information, we test filters operating with and without floor plans. Furthermore, based on statistics of the Monte Carlo error of the filter we find that 400 particles is enough to achieve reasonable estimation accuracy.

This paper is organized as follows. In section II it is described how PDR, WLAN, barometer and floor plan information are used for positioning and in section III particle filter algorithms for 2D and 3D cases are presented. Section IV covers particle smoothing for 2D positioning. In section

978-1-4673-1954-6/12/$31.00

V the test results are presented and section VI concludes the paper.

Notations: $t_3(\mathbf{m}, \mathrm{P})$ refers to the non-central scaled Student's t distribution with 3 degrees of freedom and with mean $\mathbf{m}$ and covariance matrix P (shape matrix $\frac{1}{3}\mathrm{P}$), and $t_3(\mathbf{x}|\mathbf{m}, \mathrm{P})$ refers to its probability density function (pdf) evaluated at $\mathbf{x}$.

II. Used measurements

A. Pedestrian dead reckoning

Our 2D filter implements an inertial pedestrian dead reckoning (PDR) system using MEMS-based (Micro Electro Mechanical Systems) accelerometer, gyroscope and barometer data. In many sources, the sensor unit is assumed to be attached to the user's shoe. However, this is not practical in many applications, in which the user is a private person, who cannot be assumed to attach any extra devices to his or her clothing. For example, if the positioning system is designed to function in a mobile device, very little can be assumed of the sensors' placements. Therefore, in the test cases presented in this paper the sensor unit is held in the hand. It is only assumed that the sensor unit's orientation with respect to the user does not change or the orientation can be estimated continuously.

Since the sensor unit is hand-held, the accelerometer data are much noisier than for shoe-mounted systems. Thus, our PDR algorithm is not based on computing displacements by double integration but on detecting the footsteps and estimating each step's length. Steps are detected based on the norm of the low-pass filtered acceleration as suggested by Leppäkoski et al. in [2], and the change of the user's heading during the step is inferred from the average angular velocity in the 2D plane indicated by the gyro. The authors of [2] use the step duration to infer the step length, and other methods have also been proposed [12]. However, we found that this method is not reliable enough, and in this paper the step length is estimated on the higher layer by the particle filter.

For example, [4] uses magnetometer (i.e. compass) data to obtain absolute heading information, but we do not use it because its reliability is questionable in many indoor environments [13]. Instead, the absolute heading is inferred only by the particle methods.

The barometer measures the air pressure from which the changes in altitude can be inferred accurately within short time intervals. However, the reliability may suffer from changes in the indoor air conditioning systems and in longer time periods from changes in the outdoor air pressure. Thus, continual recalibration of the mapping between barometric pressure and altitude is required. Barometer information is used by the 3D particle filter to detect floor changes.

B. Positioning with WLAN measurements

Besides floor plans, WLANs are the only source of absolute position information in the system, and WLAN is the only information channel that can produce a static snapshot estimate of the position. The position information contained by the WLANs is based on training data that must be collected from each floor of each building beforehand. It is assumed that the collected fingerprints contain accurately known position, list of the identifiers of the heard access points (AP) and the corresponding received signal strength indicators (RSSI). Furthermore, it is assumed that the RSSI readings of different devices can be calibrated so that they are comparable and that the received signal strength (RSS) in dBm can be computed based on the RSSI [14, 15]. All the data used in this paper's tests are collected with similar devices.

There are several ways to infer the user position using WLAN signals. This paper uses only RSS, since their usage does not require any external hardware modifications, and they can be measured by the receiving terminal alone. To average out noise and to keep the number of stored parameters small, the standard logarithmic path loss model is used to model the dependency of RSS from user position:

$$P = A - 10n \log_{10}(||\mathbf{r} - \mathbf{m}||) + w_{\mathrm{PL}}, \tag{1}$$

where P is the measured RSSI level, $\mathbf{r}$ is the user's position, $\mathbf{m}$ the AP's position, and $w_{\mathrm{PL}} \sim \mathrm{N}\left(0, \sigma^2\right)$ is a normally distributed shadowing component. For the shadowing standard deviation we used the constant $6\,\mathrm{dB}$. The model parameters A and n as well as the location of the AP are estimated by the Gauss–Newton method as in [16].

The positioning algorithm is also a Gauss–Newton method presented in [16]. Since the WLAN measurements are in this paper fused with other types of position measurements in time-series, it is critical that the variance of the position estimate is estimated correctly. Therefore, the uncertainties of the path loss parameters and AP positions are also estimated and used in the positioning phase as suggested in [16]. For example [2, 5, 10] use a global uncertainty parameter for the position coordinates, but we have found this assumption to be unsatisfactory, since the achievable precision in the position domain depends heavily on the WLAN environment; it can vary significantly even within one building.

C. Floor plan information in positioning

Another means to compensate the sensor drifts and to make the estimate more accurate is to use floor plans. This paper assumes that the floor plans of every floor for every building of interest are available. A floor plan is a set of thin walls, each described by five numbers: the floor and the coordinates of the end points of the wall. Doors are modeled as gaps in the walls. Furthermore, some additional information is included such as floor heights and the locations of spaces that allow floor change, such as staircases and elevators. Information on furniture or other movable objects is not used due to its changeable nature.

Inaccuracies must be taken into account. There may be errors in the maps, such as missing or nonexistent doors or walls. Nonexistent walls and missing doors in the maps can be coped with by allowing a small probability of going through a wall [10]. This is a trade-off between accuracy and robustness of the probabilistic model, since some information is unavoidably lost if wall penetration is allowed.

This paper assumes that the building is known. Building detection is left for future research.

III. Particle filtering

A. State model

A particle filter is a Monte Carlo algorithm that approximates the posterior distributions $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ provided that certain Markov assumptions hold and the probability distributions $p(\mathbf{x}_0)$, $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ and $p(\mathbf{y}_k|\mathbf{x}_k)$ are known and their density values are computable for each time index k. The components of the vector $\mathbf{x}$ are the state variables, and the vector $\mathbf{y}$ contains the measurements. No assumptions of linearity or Gaussianity have to be made.

In this article, the state consists of 2-dimensional position $\mathbf{r}$, heading α, step length ℓ and possibly altitude z:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{r}_k \\ \alpha_k \\ \ell_k \\ z_k \end{bmatrix}$$

The PDR output that contains step detection, heading change and altitude change is involved in the state transition model. The process noise is chosen to be Student's t distributed, which choice is discussed later in this section. Thus, the state transition density for each step index k is

$$\begin{aligned} & p(\mathbf{x}_{k+1}|\mathbf{x}_k) \\ =& p(\mathbf{r}_{k+1}, \alpha_{k+1}, \ell_{k+1}, z_{k+1}|\mathbf{r}_k, \alpha_k, \ell_k, z_k) \\ =& p(\alpha_{k+1}, \ell_{k+1}, z_{k+1}|\mathbf{r}_k, \alpha_k, \ell_k, z_k) \\ & \cdot p(\mathbf{r}_{k+1}|\alpha_{k+1}, \ell_{k+1}, z_{k+1}, \mathbf{r}_k, \alpha_k, \ell_k, z_k) \\ =& t_3\left(\begin{bmatrix} \alpha_{k+1} \\ \ell_{k+1} \\ z_{k+1} \end{bmatrix} \middle| \begin{bmatrix} \alpha_k + \Delta_k \\ \ell_k \\ z_k + b_k \end{bmatrix}, \mathrm{Q}_k \right) \\ & \cdot t_3\left(\mathbf{r}_{k+1}|\mathbf{r}_k + \ell_{k+1} \cdot \begin{bmatrix} \cos(\alpha_{k+1}) \\ \sin(\alpha_{k+1}) \end{bmatrix}, \mathrm{P}_k \right) \end{aligned} \tag{2}$$

where Δ is the heading change indicated by the PDR and b is the altitude change indicated by the barometer. The footstep length's distribution is restricted *a priori* so that it is always a sensible step length. The process noise covariance matrix for position is $\mathrm{P}_k = \sigma_{x,y}^2 \cdot \mathrm{I}$, where $\sigma_{x,y}$ is a configuration parameter. The process noise covariance matrix for the rest of the state parameters α, ℓ and z is assumed to be diagonal, so it is

$$\mathrm{Q}_k = \begin{bmatrix} \Delta t_k \cdot \sigma_\Delta^2 & & \mathrm{O} \\ & \sigma_\ell^2 & \\ \mathrm{O} & & \sigma_b^2 \end{bmatrix},$$

with $\sigma_b = 0$ in the 2-dimensional method with known altitude.

The process noise variances are configuration parameters that are set off-line. We emphasize the significance of the variance parameters, since they determine the magnitude of the smoothing effect of the filter: if this is too low, useful time-series information is neglected, but if the filter is over-smoothing, the particle cloud may fail to cover the whole interesting state-space area. A filter with too small process noise variance might e.g. not find a correct door or narrow corridor when a turn happens after a long time since the last WLAN measurement. We found that it is especially recommendable to use somewhat larger noise variances than the actual accuracy of the PDR would indicate, if there is a danger that WLAN measurements contain outliers with respect to the assumed measurement uncertainties. If an outlying WLAN measurement biases the particle cloud, the state-space area of actual interest might become a low-probability area that is inadequately or not at all covered by the particles. Large process variances and the heavy-tailed Student's t distribution then increase the coverage of the particle cloud more rapidly. In the literature, this *ad-hoc* uncertainty increase is called jittering or roughening [17].

One aspect in the process noise variance is the ratio of the position noise to the noise of the motion model variables heading and step length. Because only position is measured directly, too large $\sigma_{x,y}$ with respect to σ_α and σ_ℓ results in incapability of estimating heading and step length. Furthermore, too small ratio results in over-learning heading and step length; their estimates tend to become biased in order to correct a bias in position estimate, which eventually leads to overcorrecting the position estimate, i.e. correcting the position estimate beyond what is needed. Therefore, more sophisticated PDR solutions with as little noise as possible are of great value.

The state estimate is corrected using two kinds of measurement models: floor plan and WLAN. Based on the floor plan information, the probability of the wall penetrating transitions is zero or at least small. Formally, the floor plan measurement model is expressed as

$$\mathbb{P}(C_k|\mathbf{r}_k, \mathbf{r}_{k-1}) = \epsilon, \tag{3}$$

where

$$C_k = \text{“There is a wall in the map that crosses the } k\text{th step”}.$$

Thus, the number ϵ is a permeability coefficient which models the probability that in the map there is a step-crossing wall in the map is actually nonexistent. The inequality $0 \leq \epsilon \ll 1$ should hold. In our tests, the particles that enter inaccessible areas as well as the particles that move out of the building are given weight zero.

Having impermeable walls i.e. having small ϵ tends to introduce some inefficiency to the system, since lots of particles are eliminated regularly, these particles thus becoming useless until the next resampling. Therefore, [7] proposes a retry procedure: if the generated particle collides with wall, it is regenerated several times before deletion. However, one must be careful with this. Firstly, wall collision checking is computationally the most expensive part of the filter. Secondly, retry changes the statistical model: If step length is being estimated by the particles, retry increases the probability of short steps in the state transition model. If step length is taken directly from the PDR as in [7], it reduces the weight of PDR step length measurement and gives more probability to a short step. Instead of this, we consider wall collisions to indicate that the particles original state was improbable and the particle deserves to be given a lower weight. Retry could only be applied as a heuristic robustness-increasing method in limited cases where only a few particles are in the interesting state-space area, but this is not considered further in this paper.

The choice of parameter ϵ has significant influence on the filter behaviour: Allowing wall penetration, i.e. setting $\epsilon > 0$, adds robustness against errors in the map and, for example, against inconsistent WLAN measurement which tend to result in a too small particle cloud that does not cover the true position. On the other hand, compared to the model $\epsilon = 0$ somewhat less map information is used, and if the particle cloud is inconsistent, the choice $\epsilon > 0$ may allow the whole particle cloud to penetrate a wall without any effect on the weighting. If resampling is done when part of the cloud has penetrated a wall and the rest will penetrate at the next step, there is a risk that the particle estimate starts to lag from the true estimate. This phenomenon is due to the Monte Carlo approximation of the distribution, and techniques for avoiding it are a topic for future research.

The true likelihood of one WLAN measurement is not Gaussian, but it is approximated with a Gaussian by the Gauss–Newton method as described in [16]. Thus, the used WLAN measurement model is

$$p(\mathbf{y}_k|\mathbf{r}_k, \alpha_k, \ell_k, z_k) \overset{\mathbf{r}_k}{\propto} \mathrm{N}\left(\mathbf{r}_k|\mu_k(\mathbf{y}_k), \Sigma_k(\mathbf{y}_k)\right). \tag{4}$$

In our method the measurement covariance matrix $\Sigma_k(\mathbf{y}_k)$ is not a configuration parameter but it is returned by the Gauss–Newton method. For fusion with other measurement information, it is crucial that the covariance matrices are in accordance with the real precision of the WLAN measurements.

B. Particle filter algorithm

A particle filter approximates the presented model's posterior distributions with a set of weighted particles $\{(\mathbf{x}_k^i, W_k^i)|i \in \{1, \ldots, N\}\}$ that are random realizations of the state space [1, 18]. The random samples $\mathbf{x}_k^i$ follow the importance distribution, also known as the proposal distribution, and the weights W_k^i, also known as importance weights, make the sample approximate the true posterior in the sense that every moment of the distribution are approximated by weighted sums over the particle set:

$$\mathrm{E}(\mathbf{g}(\mathbf{x}_k)|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N} W_k^i \cdot \mathbf{g}(\mathbf{x}_k^i)$$

where $\mathbf{g}$ is an almost arbitrary Borel measurable function and N is the number of particles. This is conventionally denoted by the Dirac delta notation

$$p(\mathbf{x}_k|\mathbf{y}_{1:k}) \approx \sum_{i=1}^{N} W_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i).$$

Thus, the posterior is approximated by a set of sharp distribution peaks.

In the initial phase the particle values are generated from the initial prior $p(\mathbf{x}_0)$ with equal weights. The particle filter is updated whenever a footstep is detected or a WLAN measurement is received. As for the Kalman-type filters, one update consists of two stages: the prediction stage, in which the particles of new time instant are generated from the prespecified proposal distributions $q(\mathbf{x}_k|\mathbf{x}_{1:k-1}^i, \mathbf{y}_{1:k})$, and the update stage, in which the measurements are used to update the weights of the particles. Given the previous timestep's weights W_{k-1}^i, the current time instant's unnormalized weights $\tilde{W}_k^i$ are obtained using the formula

$$\tilde{W}_k^i = p(\mathbf{y}_k|\mathbf{x}_k^i)\mathbb{P}(C_k|\mathbf{x}_{k-1:k}^i) \cdot \frac{p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{1:k-1}^i, \mathbf{y}_{1:k})} \cdot W_{k-1}^i, \tag{5}$$

in which the first terms represent the measurement likelihood and the middle term the ratio of state model and proposal distribution. The normalization factors of the pdf's $p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)$ and $q(\mathbf{x}_k^i|\mathbf{x}_{1:k-1}^i, \mathbf{y}_{1:k})$ are not required, since the normalization constants are often well approximated by

$$W_k^i = \frac{\tilde{W}_k^i}{\sum_{i=1}^{N} \tilde{W}_k^i} \tag{6}$$

[1, Ch. 2.5.2].

The choice of the proposal distribution is a crucial ingredient of any particle filter. A rule of thumb is that the proposal distribution should be as close to the final posterior as possible. In this paper the state transition distribution $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$ is used as a proposal distribution.

A third stage is required by any particle filter to avoid all the weight concentrating to one particle: resampling [18, Ch. 3.3]. At the resampling stage the particles' weights are equalized by sampling with replacement N new particles from the old particle set with the old particle weights as probabilities. One drawback of this resampling method is the danger of sample impoverishment, which means that most of the particles are resampled to one or a few locations, which may lead to inadequate coverage of the interesting state space areas [18]. A conventional method for avoiding this is to resample only at some time steps. A standard approach is to resample only when the effective number of particles (ENP), $N_{\mathrm{eff},k} = 1/\sum_{i=1}^{N}(W_k^i)^2$, goes below some threshold [18, Ch. 3.3]. Since resampling increases the Monte Carlo variance of the estimate, the estimate reported before the resampling, and with step detection the resampling is only performed when a footstep is detected and the ENP condition is fulfilled.

With the model of this paper, the prediction stage updates the particle values with PDR readings and randomly generated noise modifying the weights of wall-crossing particles, and the update stage corrects the particle weights based on the WLAN measurement. In case a WLAN scan is not made at some time instant, the update stage is simply omitted. The detailed algorithm description is present in Algorithm 1.

C. Initialization and divergence monitoring

To ensure that the particle filter estimate converges to the true posterior in a reasonable number of time steps with a reasonable number of particles, it is crucial that the initial prior

Algorithm 1 Particle filter for 2D indoor positioning

```
1) For each $i \in \{1, \ldots, N\}$ set $W_0^i := \frac{1}{N}$ and generate $\mathbf{x}_0^i \leftarrow p(\mathbf{x}_0)$. Set the time index $k := 1$.
2) Set $\tilde{W}_k^i := W_k^i$. If no footstep is detected at time window index $k$, go to Phase 5. Otherwise, if $\frac{1}{\sum_{i=1}^{N} (W_{k-1}^i)^2} < N_{\text{eff,lim}}$, perform resampling.
3) For each $i \in \{1, \ldots, N\}$ generate
       $\begin{bmatrix} \alpha_k^i \\ \ell_k^i \end{bmatrix} \leftarrow t_3\left( \begin{bmatrix} \alpha_{k-1}^i + \Delta_{k-1} \\ \ell_{k-1}^i \end{bmatrix}, \mathrm{Q}_{k-1} \right)$
       $\mathbf{r}_k^i \leftarrow t_3\left( \mathbf{r}_{k-1}^i + \ell_k^i \cdot \begin{bmatrix} \cos(\alpha_k^i) \\ \sin(\alpha_k^i) \end{bmatrix}, \mathrm{P}_{k-1} \right).$
4) Set
       $\tilde{W}_k^{i^*} := \frac{\epsilon}{1-\epsilon} \tilde{W}_{k-1}^{i^*}$
   for all $i^*$ such that there is a wall between $\mathbf{r}_{k-1}$ and $\mathbf{r}_k$.
5) If no WLAN measurement is obtained at time index $k$, go to Phase 6. Otherwise, perform the Gauss–Newton algorithm to obtain the WLAN position's mean $\mu_k$ and covariance matrix $\Sigma_k$. Set
       $\tilde{W}_k^i := \mathrm{N}\left(\mu_k | \mathbf{r}_k^i, \Sigma_k\right) \cdot \tilde{W}_{k-1}^i$
   for each $i \in \{1, \ldots, N\}$.
6) Normalize the weights by
       $W_k^i := \frac{\tilde{W}_k^i}{\sum_{i=1}^{N} \tilde{W}_k^i}.$
   Report the mean and covariance matrix for position
       $\hat{\mu}_k := \sum_{i=1}^{N} W_k^i \cdot \mathbf{r}_k^i, \ \hat{\Sigma}_k := \sum_{i=1}^{N} W_k^i \cdot (\mathbf{r}_k^i - \hat{\mu}_k)(\mathbf{r}_k^i - \hat{\mu}_k)^{\mathrm{T}}$
   If the positioning ends, stop. Otherwise, set $k := k + 1$ and go to Phase 2.
```

is accurate enough for each component of the state, i.e. it does not have a significant bias and the variance is small. Otherwise especially the convergence of heading and step length may be very slow, since they are not measured directly, but only through position. Especially if WLAN accuracy is low, there is also a danger that large open areas are overweighted, since inaccurate particles are less likely to be eliminated by wall collisions there than in more confined spaces.

The accuracy of the initial prior and thus the applied particle initialization method depends on the scenario. If the user comes from outdoors and has been using GPS or some other accurate positioning method, there is likely to be a feasible prior distribution for position and heading and even for step length. If, however, the positioning device is switched on indoors without any history of positioning data, there is no prior information.

One initialization scenario is that the particles have got stuck in an area that does not fit well with the WLAN measurements. Due to the "hard" nature of wall constraints even a relatively small bias in position may result in getting stuck, and the recovery may be slow or almost impossible. Allowing the particles to penetrate walls, i.e. setting $\epsilon > 0$ in Eq. (3) may help but does not completely resolve the issue, since wall penetrations delay the estimate, or in the case of multimodal cloud they may result in overweighting large open areas. Since the standard form of particle filter does not perform a global state space search, recovery from filter divergence requires that all or some of the particles are re-initialized.

Our solution to both the initialization problems is to maintain a robust light-weight fallback filter in the background. The fallback filter should perform global state space search and be independent of the floor plan constraints. For this we use the Kalman filter (KF) described by Raitoharju et al in [19]. This algorithm has 2D position and 2D step vector in the state, and the step detection and heading given by the PDR are fed in at the prediction stage and WLAN measurements at the update stage. Thus, the algorithm estimates simultaneously position, heading and footstep length using a linear state model. To make the algorithm more robust against outlying WLAN estimates, the measurements are assumed to have Student's t noise distribution. This model can be solved approximatively using a Variational Bayes-based version of KF [20].

In the case of unknown initial state the fallback KF is initialized with the mean of position at an arbitrary position such as the middle point of the building and large variance. As soon as the KF estimate has converged, i.e. its variance for the step is small enough, the KF estimate is used for initializing the particles. Generating the particles from the normal distribution defined by the KF output is straightforward.

After the first initialization and in the case of known initial state, the fallback KF is used for monitoring the quality of the particle cloud, i.e. for checking the re-initialization criterion and for estimating the distribution of the re-initialized particles. At re-initialization, it might be advantageous to re-initialize only $100 \cdot \lambda\,\%$ of the particle cloud and sample the rest from the old particle set, λ being a configuration parameter for which $0 \ll \lambda \leq 1$.

The criterion for when to re-initialize the existing particles needs some consideration. In [9], re-initialization is done when the Kullback–Leibler divergence (KLd) between the normalized measurement likelihood and the prior implied by the particles is large. However, we did not find this a suitable criterion, since the KLd measures also the difference in the uncertainties of the distributions, and the accuracy of the measurement is never dependent on the accuracy of prior. In [7] the re-initialization decision is based on the range measurements' deviations from the least deviating particle's prediction, and in [21] the unnormalized particle likelihoods are monitored. However, there might be outliers among the WLAN measurements, and in such a case the re-initialization might result in significant information losses. Therefore, we do not use the static WLAN measurement but the more robust KF estimate, which also considers the heading and step length of the particles. If the best particle is improbable given the

4-dimensional KF distribution, the particles are re-initialized based on the KF estimate. "Best" refers here to the particle that is closest to the KF estimate of all positively weighted particles. This algorithm is described in Algorithm 2.

Algorithm 2 Re-initialization of particle filter

1) After Phase 5 of Algorithm 1, feed the Kalman filter (KF) with PDR and WLAN estimate to get the KF estimate (μ_k^K, Σ_k^K).
2) Compute the step vector for each positively-weighted particle $i \in \{i | \tilde{W}_k^i > 0\}$

$$\mathbf{s}_k^i := \ell_k^i \cdot \begin{bmatrix} \cos(\alpha_k^i) \\ \sin(\alpha_k^i) \end{bmatrix}.$$

and compute the KF deviances

$$d_k^i := \left(\begin{bmatrix} \mathbf{r}_k^i \\ \mathbf{s}_k^i \end{bmatrix} - \mu_k^K \right)^{\mathrm{T}} (\Sigma_k^K)^{-1} \left(\begin{bmatrix} \mathbf{r}_k^i \\ \mathbf{s}_k^i \end{bmatrix} - \mu_k^K \right).$$

Select the smallest KF deviance

$$d_k^* := \min_{i \in \{i | \tilde{W}_k^i > 0\}} (d_k^i).$$

3) If d_k^* exceeds the threshold value, sample $\mathrm{round}(\lambda N)$ new particles from the KF distribution and sample $N - \mathrm{round}(\lambda N)$ particles from the old weighted particle set with replacement. Give all the sampled particles equal weights. Go to Phase 6 of Algorithm 1.

D. 3D indoor positioning

Currently, an active field of indoor positioning research is floor estimation. For the end user, it is important to see the correct floor's map, and the choices of floor plan and WLAN models influence 2D positioning performance. A calibrated barometer is accurate in short-term positioning, but it drifts over time due to changes in the air pressure. Furthermore, it may also be biased by air conditioning systems.

Static floor estimation based on WLAN fingerprinting has been investigated by [22–24], among others. Article [25] also incorporates barometer information, and calibrates the barometer with WLAN. Results in multifloor particle filter positioning have been presented at least by [6, 26].

Provided that the barometer is in use, the extension of the presented particle filter algorithm into 3D position space is straightforward in principle. Given the means of the barometric pressures over the previous and latest time intervals ρ_{k-1} and ρ_k, the mean of the predicted altitude change in Eq. (2) is

$$b_k = \frac{RT}{gM} \ln \left(\frac{\rho_k}{\rho_{k-1}} \right), \tag{7}$$

where R is the molar gas constant, T is the temperature for which 293 K was used in our tests, g is the acceleration of gravity, and M is the molar mass of air [27, Ch. 2.1]. Provided that the floor elevations are known, a particle's floor is always implied by the particle's altitude, so the probability of each floor can be computed by summing the weights of the particles that are located in the floor. The reported position estimate is computed using only the particles of the most probable floor.

The WLAN measurements are used straightforwardly in the 3D method provided that the floor is always known in the learning phase: the path loss parameters are learnt separately for each floor, and in the positioning phase the used parameters are chosen separately for each particle according to the floor of the particle. Because the Gauss–Newton method returns only the mean and covariance matrix for position, this method normalizes the likelihood for each floor with respect to user position. However, the normalization constants are different for different floors, so some of the absolute floor information is neglected by this method. This is another topic for future research.

The floor plans provide extra information for the floor change cases, if staircases, elevators and other spaces that connect different floors are labeled there. The floor forms a similar kind of map restriction as the walls in planar positioning: there are "doors" that connect two floors, and otherwise the floor change is only possible with small probability ϵ_f, which is the probability that the used connector is missing from the map.

Similarly to the 2D setup, there is a possibility that the particle filter diverges, that is, it cannot find the true floor e.g. due to failure in finding the connector. Moreover, if the whole particle cloud changes floor without finding a connector, the weighting is not affected when $\epsilon_f > 0$, except that if the cloud is resampled during the penetration, the altitude estimate starts to lag. There could be a connector-independent fallback filtering algorithm for the altitude dimension also, and the estimate of this algorithm could be used to measure the adequacy of the particle cloud coverage and possibly for multifloor re-initialization. However, this idea was not implemented for this paper, but it is left for future research.

IV. Particle smoothing

A. Algorithm derivation

The goal of Bayesian smoothing is to compute the marginal smoothed distributions $p(\mathbf{x}_k | \mathbf{y}_{1:T})$ where $T > k$. Compared with Bayesian filtering also future measurements $\mathbf{y}_{k+1}, ..., \mathbf{y}_T$ are used to estimate the state $\mathbf{x}_k$, and hence it is possible to achieve better estimation accuracy. A computationally efficient way to do particle smoothing is to do it while filtering and keeping in memory the particle trajectories. This can be done by computing first the joint distribution for the whole time interval of interest $k \in \{0, \ldots, T\}$

$$p(\mathbf{x}_{0:T} | \mathbf{y}_{1:T}) \approx \sum_{i=1}^{N} W_T^i \delta(\mathbf{x}_{0:T} - \mathbf{x}_{0:T}^i), \tag{8}$$

and then using the latest weights W_T^i and particles from each time step to approximate the smoothed marginals

$$p(\mathbf{x}_k | \mathbf{y}_{1:T}) \approx \sum_{i=1}^{N} W_T^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \tag{9}$$

[28, p. 662, 674]. This type of smoothing is used in [4] where the Backtracking Particle Filter (BPF) is introduced.

BPF uses the latest weights to correct the estimate of the past state $\mathbf{x}_{k-m}$. In [4, p. 210, 212] the value of m is established empirically.

A problem with this smoothing method is that it produces degenerate approximations of smoothed distributions at times $k \ll T$ [28, p. 698], [29]. The smoothing method used in this paper is the Forward Filtering-Backward Smoothing (FFBS) which does not suffer that much from the degeneracy problem. In the FFBS the reweighing of the particles is based on the equation

$$p(\mathbf{x}_k|\mathbf{y}_{1:T}) = p(\mathbf{x}_k|\mathbf{y}_{1:k}) \int \frac{p(\mathbf{x}_{k+1}|\mathbf{x}_k)}{p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k})} p(\mathbf{x}_{k+1}|\mathbf{y}_{1:T}) d\mathbf{x}_{k+1}, \tag{10}$$

which implies the smoothing formula for the weights

$$W_{k|T}^i = W_k^i \cdot \left[\sum_{j=1}^N W_{k+1|T}^j \frac{p(\mathbf{x}_{k+1}^j|\mathbf{x}_k^i)}{\sum_{l=1}^N W_k^l p(\mathbf{x}_{k+1}^j|\mathbf{x}_k^l)} \right], \tag{11}$$

where W_k^i are the filtering weights [11, p. 204-205]. In a practical FFBS implementation particle filtering is done first, and then new weights are computed by (11). The smoothed distributions are then represented by the new weights and the particles from the filtering phase. A disadvantage of FFBS is that it requires $O(N^2T)$ operations to approximate $p(\mathbf{x}_k|\mathbf{y}_{1:T})$ [28, p. 700]. The algorithm for FFBS is given in Algorithm 3.

Algorithm 3 Forward Filtering-Backward Smoothing

```
1) Perform PF with Algorithm 1 and store particles $\mathbf{x}_t^i$ and
   weights $W_t^i$ for each time window index $k \in \{1, \ldots, T\}$.
2) For time window index $T$, report the filter estimate. Set
   $k := T - 1$ and $W_{T|T}^i := W_T^i$.
3) Compute new weight for each particle $\mathbf{x}_k^i$ by
   $W_{k|T}^i := W_k^i \cdot \left[ \sum_{j=1}^N W_{k+1|T}^j \frac{p(\mathbf{x}_{k+1}^j|\mathbf{x}_k^i)}{\sum_{l=1}^N W_k^l p(\mathbf{x}_{k+1}^j|\mathbf{x}_k^l)} \right]$
4) Report the mean and covariance matrix for position
   $\hat{\mu}_k := \sum_{i=1}^N W_{k|T}^i \mathbf{r}_k^i,\ \hat{\Sigma}_k := \sum_{i=1}^N W_{k|T}^i (\mathbf{r}_k^i - \hat{\mu}_k)(\mathbf{r}_k^i - \hat{\mu}_k)^{\mathrm{T}}$
   If the positioning ends, stop. Otherwise, set $k := k - 1$
   and go to Phase 3.
```

B. Particle smoother for indoor positioning

FFBS is implemented and tested in 2D indoor positioning scenarios. A fixed-interval smoothing approach is presented using the whole time series as the interval, but fixed-lag smoothing would also be possible using the same formulas. The map information is not used in the smoothing phase in order to make the computations lighter. The map information is, however, taken into account in the filter weights W_k^i.

The state model of (2) can be written as a stochastic difference equation

$$\mathbf{x}_{k+1} = \mathbf{f}_k(\mathbf{x}_k, \mathbf{w}_k), \tag{12}$$

where $\mathbf{w}_k \sim t_3\left(\mathbf{0}, \begin{bmatrix} \mathrm{P}_k & \mathrm{O} \\ \mathrm{O} & \mathrm{Q}_k \end{bmatrix}\right)$ is the process noise and $\mathbf{f}_k$ is the state transition function. For computing the state transition densities $p(\mathbf{x}_{k+1}|\mathbf{x}_k)$, the state model is linearized by

$$\begin{aligned}
&\mathbf{f}_k(\mathbf{x}_k, \mathbf{w}_k) \\
&= \begin{bmatrix} r_{x,k} + (\ell_k + w_{4,k})\cos(\alpha_k + \Delta_k + w_{3,k}) + w_{1,k} \\ r_{y,k} + (\ell_k + w_{4,k})\sin(\alpha_k + \Delta_k + w_{3,k}) + w_{2,k} \\ \alpha_k + \Delta_k + w_{3,k} \\ \ell_k + w_{4,k} \end{bmatrix} \\
&\approx \begin{bmatrix} r_{x,k} + \ell_k \cos(\alpha_k + \Delta_k) \\ r_{y,k} + \ell_k \sin(\alpha_k + \Delta_k) \\ \alpha_k + \Delta_k \\ \ell_k \end{bmatrix} \\
&\quad + \begin{bmatrix} 1 & 0 & -\ell_k \sin(\alpha_k + \Delta_k) & \cos(\alpha_k + \Delta_k) \\ 0 & 1 & \ell_k \cos(\alpha_k + \Delta_k) & \sin(\alpha_k + \Delta_k) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_{1,k} \\ w_{2,k} \\ w_{3,k} \\ w_{4,k} \end{bmatrix} \\
&\triangleq \mathbf{f}_k^*(\mathbf{x}_k) + \mathrm{C}_k \mathbf{w}_k.
\end{aligned} \tag{13}$$

Note that $\mathrm{C}_k\mathbf{w}_k \sim t_3\left(\mathbf{0}, \mathrm{C}_k \begin{bmatrix} \mathrm{P}_k & \mathrm{O} \\ \mathrm{O} & \mathrm{Q}_k \end{bmatrix} \mathrm{C}_k^{\mathrm{T}}\right)$, because if $\mathbf{w} \sim t_v(\mu, \Sigma)$ and C is invertible, $\mathrm{C}\mathbf{w} \sim t_v(\mathrm{C}\mu, \mathrm{C}\Sigma\mathrm{C}^{\mathrm{T}})$ holds [30, Ch. 1.9].

V. Testing

A. Equipment and environment

Particle filter and FFBS particle smoother are tested in Tampere University of Technology campus building Tietotalo. The presented test tracks are located in corridors that are surrounded by offices. Only the existing WLAN architecture is used. The inertial sensor unit is Xsens MTx. Acer Iconia Tab W500 tablet PC with Windows 7 OS is used to log the WLAN measurements in the learning and positioning phase. The used indoor maps are HERE Destination Maps. The reference locations are set manually by the user by tapping the floor plan figure on the tablet's screen. Positioning algorithms are computed with MATLAB using MEX-files in the most critical parts of the code to speed up the computation.

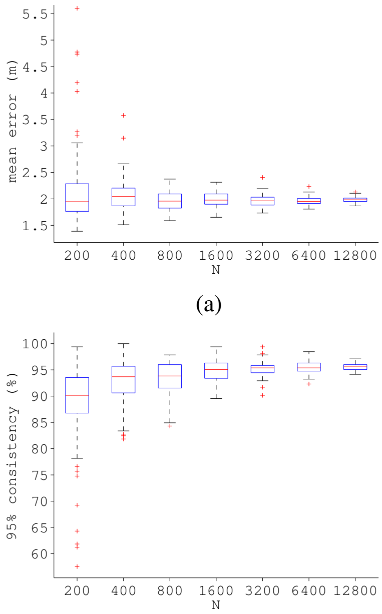
The results of two positioning tracks are shown in this paper. The first one is in one floor (the 2D track), and the other one contains a floor change from floor 2 to floor 3 (the 3D track). The WLAN Learning data have been collected from each floor of the five-story building. At the time of the test track collection, the learning data were about three months old. Floor 2 has path loss models for 103 APs and floor 3 for 126 APs. Typically, 15–40 APs are observed at each scan, 25 on average. During the data collection, the inertial sensor unit was held in the hand avoiding rotations with respect to the user. WLANs were scanned every ten seconds. The floor change in the 3D track was made using stairs.

The solvers are given the correct initial position and heading with variances $(2\,\mathrm{m})^2 \cdot \mathrm{I}_{2\times 2}$ and $(10°)^2$. The step length prior is $\mathrm{N}\left(0.7\,\mathrm{m}, (0.015\,\mathrm{m})^2\right)$. In the 3D test, the initial floor is assumed to be known, and the prior for altitude is the normal distribution with mean in the correct floor's altitude and variance depending on the floor's elevation.

B. Results and discussion

Figure 1 shows box plots of the mean errors of 100 particle filter runs on the 2D track with different numbers of particles. From the plot we infer that 400 particles is enough to achieve the best possible median performance. However, if the reliability of the filter is to be improved, the number can be increased. The convergence rate of Monte Carlo integration is known to be $O(\sqrt{N})$ [18, Ch. 3.1]. Adding dimensions, such as altitude, to the state tends to slow up convergence.

Our measurement device is capable of processing the 400-particle filter in real time. Our implementation is not highly optimized. However, the set of walls has been divided into groups so that wall crossing is checked only with the walls that are close to the moving particle. Thus, the complexity of the bottleneck phase does not depend on the total number of walls and floors.



(a)

(b)

Figure 1. Boxplots of the mean errors (1a) and 95 % consistencies (1b) of 100 particle filter runs on the 2D track varying the particle number N

For the 2D track, the error statistics for filters and smoother are presented in Table I. Particle filter (PF) and smoother (FFBS) are the only methods using floor plans. The reference methods are static WLAN, the Kalman filter (KF) described in Section III-C, and particle filter without map information. The errors of the particle filters and FFBS are averages of the error statistics of 100 Monte Carlo simulations with 400 particles. The error statistics are empirical mean, median and 95 % quantile of the weighted particle mean's errors with respect to the reference trajectory. Note that the error of the static WLAN algorithm is computed only at the time instants when WLAN measurement is received. For all the other algorithms, the error is computed with 0.5 s interval. The 95 % consistency is determined using the Gaussian consistency test [31, p. 235]: A solver is deemed to be consistent at a certain time step if the true position is within the 95 %-ellipse of the posterior distribution, assuming normality of the posterior. The closer this number is to 95 %, the better the weighted covariance matrix of the particle cloud corresponds to the realized error. More rigorous consistency evaluation of non-Gaussian distributions is left for future research.

TABLE I. Positioning results for static WLAN, KF, PF with and without floor plan information and FFBS on the 2D track

	mean error (m)	median error(m)	95 % quant. of errors (m)	95 % cons. (%)
static WLAN	7.5	4.7	20.7	80
Kalman filter	4.8	3.2	16.3	83
PF (no map)	5.2	3.3	17.7	83
PF	2.0	1.3	6.0	92
FFBS	1.4	0.8	4.9	61

The KF combines WLAN and PDR in time-series. Based on the error statistics of Table I, the KF reduces the error by about 25 %. PF without floor plans uses the same measurements and gives similar results with the KF, which validates the linear motion model of the KF. Incorporating the floor plans (map) to the PF yields further significant decrease in errors. Also, the estimate crosses walls more rarely, which provides a better user experience. The particle smoother FFBS decreases the errors further. The performance differences of PF and FFBS in one example case are analyzed in the following paragraph.

Estimated tracks for one Monte Carlo simulation are in Figure 2. In subfigures 3a and 3b there are snapshots of the particle cloud and particle filter estimate moving from one corridor to another. In subfigure 3a the weighted mean is penetrating through the walls but there are particles also in the corridor where the reference position is. In subfigure 3b, most of the particles are resampled in the corridor and the weighted mean also moves there. Thus, due to the future information the particles that went along the corridors are given more weight in the smoothing phase. The improvement provided by smoothing is also visible in Figure 2.

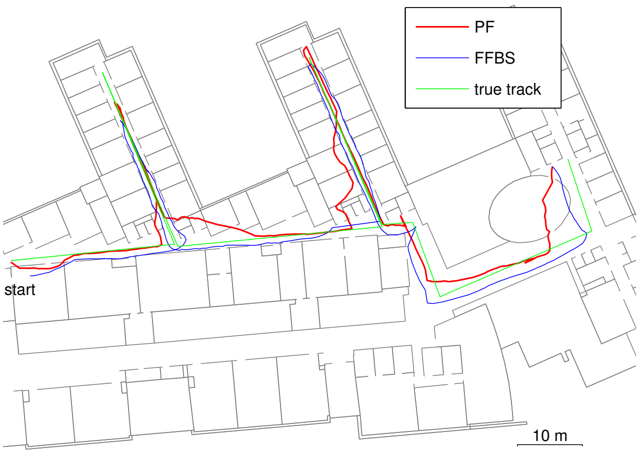


Figure 2. Estimated tracks for PF and FFBS

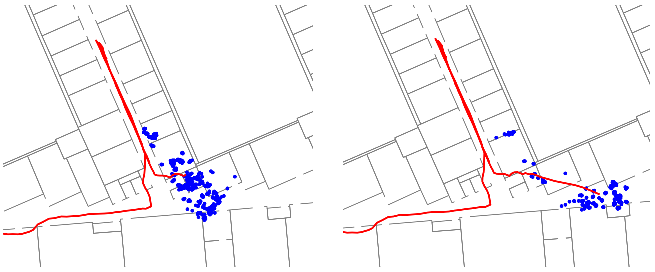
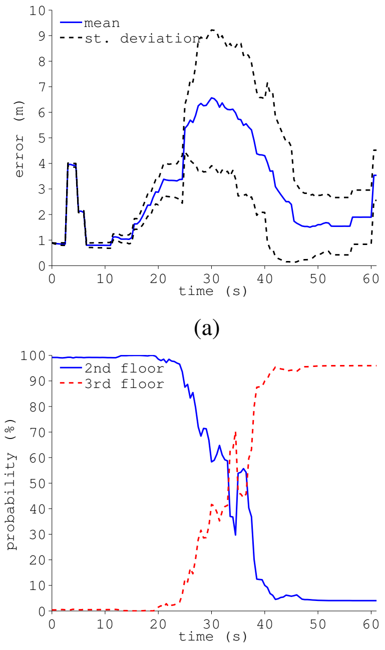

(a) (b)

Figure 3. Particle cloud moving from one corridor to another and the estimated particle filter track

Figure 4 shows the distribution of 2D error as well as the average floor probabilities estimated by the particles for the 3D track. The plots are based on 100 Monte Carlo iterations, and the initial prior is slightly biased and has somewhat more uncertainty than in the 2D case, the variances being $(5\,\mathrm{m})^2 \cdot \mathrm{I}_{2\times2}$ for position and $(20°)^2$ for heading. In this case the filter is capable of detecting the floor change and also reducing the 2D error using the floor change information. At first, the 2D error increases, because apart from the particles that are in the new floor inside a connector also those that have stayed in the old floor may survive. However, the first WLAN measurement after the floor change fits better with the particles in the new floor, so the estimate tends to converge to the correct subcloud.



(a)

(b)

Figure 4. Distribution for 2D error of position (4a) and average floor probabilities (4b) in the 3D track for 100 Monte Carlo simulations

VI. Conclusions

This article presented a particle filtering and smoothing algorithms for the hybrid indoor positioning problem involving measurements from accelerometer, gyroscope, barometer, WLAN and floor plans. It is also straightforward to add other measurement sources, such as occasional GNSS measurements. The particle algorithms estimate user's position, heading, footstep length and altitude. The particles can be initialized and, in the case of filter divergence, re-initialized using a light-weight fallback filter.

It was shown that floor plans provide a significant improvement to the positioning accuracy and consistency and that an adequate number of particles might be 400, which should be feasible for modern high-end mobile devices. Cases were presented where smoothing helps to improve the filter estimate. Moreover, a floor change case was presented, which showed that the filter was capable of detecting the floor change and improving also the 2D accuracy using this information.

Indoor positioning using particle methods still provides several future challenges. Footstep length estimation using inertial motion sensors will be studied: the coming method might include a user-specific constant, which could then be estimated by the particle filter. The wall and floor permeability models will be considered further to remedy the wall-penetration problem discussed in Section III-A. Building detection will be tested. Seamlessly 3D particle methods will be developed further by introducing multifloor fallback filter.

Acknowledgment

This research was funded by Nokia Inc.. The authors are grateful to Jari Syrjärinne and Lauri Wirola for support and advice. Toni Fadjukoff, Jussi Parviainen, Jussi Collin and Matti Raitoharju are thanked for their participation in development, implementation and testing of the algorithms.

References

[1] J. S. Liu, *Monte Carlo Strategies in Scientific Computing*. Springer, 2001.

[2] 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, June 2013.

[3] H. Wang, H. Lenz, A. Szabo, J. Bamberger, and U. D. Hanebeck, "WLAN-Based Pedestrian Tracking Using Particle Filters and Low-Cost MEMS Sensors," in *4th Workshop on Positioning, Navigation and Communication, 2007. WPNC '07.*, 2007, pp. 1–7.

[4] Widyawan, M. Klepal, and S. Beauregard, "A Backtracking Particle Filter for Fusing Building Plans with PDR Displacement Estimates," in *5th Workshop on Positioning, Navigation and Communication, 2008. WPNC 2008.*, 2008, pp. 207–212.

[5] F. Evennou, M. Francois, and E. Novakov, "Map-aided indoor mobile positioning system using particle filter," in *2005 IEEE Wireless Communications and Networking Conference*, vol. 4, March 2005, pp. 2490–2494.

[6] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proceedings of the 10th inter-*

national conference on Ubiquitous computing, 2008, pp. 114–123.

[7] P. Blanchart, L. He, and F. Le Gland, "Information fusion for indoor localization," in *12th International Conference on Information Fusion, 2009. FUSION '09.*, July 2009, pp. 2083–2090.

[8] J. Pinchin, C. Hide, and T. Moore, "A particle filter approach to indoor navigation using a foot mounted inertial navigation system and heuristic heading information," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, 2012, pp. 1–10.

[9] B. Turgut and R. P. Martin, "Restarting particle filters: an approach to improve the performance of dynamic indoor localization," in *IEEE Global Telecommunications Conference, 2009. GLOBECOM 2009.*, November 2009, pp. 1–7.

[10] P. Davidson, J. Collin, and J. Takala, "Application of particle filters for indoor positioning using floor plans," in *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010*, October 2010, pp. 1–4.

[11] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, pp. 197–208, 2000.

[12] W. Chen, R. Chen, Y. Chen, H. Kuusniemi, and J. Wang, "An effective pedestrian dead reckoning algorithm using a unified heading error model," in *2010 IEEE/ION Position Location and Navigation Symposium (PLANS)*, May 2010, pp. 340–347.

[13] E. R. Bachmann, X. Yun, and A. Brumfield, "Limitations of attitude estimation algorithms for inertial/magnetic sensor modules," *IEEE Robotics Automation Magazine*, vol. 14, no. 3, pp. 76–87, 2007.

[14] C. Laoudias, R. Piché, and C. G. Panayiotou, "Device signal strength self-calibration using histograms," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, 2012, pp. 1–8.

[15] M. B. Kjærgaard, "Indoor location fingerprinting with heterogeneous clients," *Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 31–43, 2011.

[16] H. Nurminen, J. Talvitie, S. Ali-Löytty, P. Müller, E.-S. Lohan, R. Piché, and M. Renfors, "Statistical path loss parameter estimation and positioning using RSS measurements in indoor wireless networks," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, November 2012, pp. 1–9.

[17] P. Fearnhead, "Sequential Monte Carlo methods in filter theory," Ph.D. dissertation, Merton College, University of Oxford, 1998.

[18] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter, Particle Filters for Tracking Applications.* Boston, London: Artech House, 2004.

[19] M. Raitoharju, H. Nurminen, and R. Piché, "A linear state model for PDR+WLAN positioning," in *Proceedings of of the Conference on Design & Architectures for Signal and Image Processing (DASIP 2013)*, in press.

[20] R. Piché, S. Särkkä, and J. Hartikainen, "Recursive outlier-robust filtering and smoothing for nonlinear systems using the multivariate Student-t distribution," in *2012 IEEE International Workshop on Machine Learning for Signal Processing*, September 2012.

[21] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, July 2010.

[22] F. Alsehly, T. Arslan, and Z. Sevak, "Indoor positioning with floor determination in multi story buildings," in *2011 International Conference on Indoor Positioning and Indoor Navigation (IPIN2011)*, 2011, pp. 1–7.

[23] H.-H. Liu and Y.-N. Yang, "Wifi-Based Indoor Positioning for Multi-Floor Environment," in *TENCON 2011*, 2011, pp. 597–601.

[24] S. Gansemer, S. Hakobyan, S. Püschel, and U. Großmann, "3D WLAN indoor positioning in multi-storey buildings," in *IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 2009. IDAACS 2009.*, 2009, pp. 669–672.

[25] H. Wang, H. Lenz, A. Szabo, U. D. Hanebeck, and J. Bamberger, "Fusion of barometric sensors, WLAN signals and building information for 3-D indoor/campus localization," in *Proceedings of International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2006)*, 2006, pp. 426–432.

[26] C. Ascher, C. Kessler, R. Weis, and G. Trommer, "Multi-floor map matching in indoor environments for mobile platforms," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN2012)*, 2012, pp. 1–8.

[27] E. U. Condon and H. Odishaw, Eds., *Handbook of Physics.* York, Pennsylvania: McGraw-Hill Book Company, 1958.

[28] A. Doucet and A. M. Johansen, "A Tutorial on Particle Filtering and Smoothing: 15 years later," in *The Oxford Handbook of Nonlinear Filtering*, D. Crisan and B. Rozovskiĭ, Eds. Oxford University Press, 2011, ch. 24, pp. 656–704.

[29] P. Fearnhead, D. Wyncoll, and J. Tawn, "A sequential smoothing algorithm with linear computational cost," *Biometrika*, vol. 97, no. 2, pp. 447–464, 2010.

[30] S. Kotz and S. Nadarajah, *Multivariate t Distributions and Their Applications.* Cambridge University Press, 2001.

[31] Y. Bar-Shalom, R. X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation, Theory Algorithms and Software.* John Wiley & Sons, 2001.