Research Article

Ubiquitous and Seamless Localization: Fusing GNSS Pseudoranges and WLAN Signal Strengths

Philipp Richter and Manuel Toledano-Ayala

Universidad Autónoma de Querétaro, Santiago de Querétaro, QRO, Mexico

Correspondence should be addressed to Philipp Richter; philipp.richter@uaq.mx

Received 10 December 2016; Revised 24 March 2017; Accepted 10 April 2017; Published 28 June 2017

Academic Editor: Olivier Julien

Copyright © 2017 Philipp Richter and Manuel Toledano-Ayala. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Ubiquitous global positioning is not feasible by GNSS alone, as it lacks accurate position fixes in dense urban centres and indoors. Hybrid positioning methods have been developed to aid GNSS in those environments. Fingerprinting localization in wireless local area networks (WLANs) is a promising aiding system because of its availability, accuracy, and error mechanisms opposed to that of GNSS. This article presents a low-cost approach to ubiquitous, seamless positioning based on a particle filter integrating GNSS pseudoranges and WLAN received signal strength indicators (RSSIs). To achieve accurate location estimates indoors/outdoors and in the transition zones, appropriate likelihood functions are essential as they determine the influence of each sensor information on the position estimate. We model the spatial RSSI distributions with Gaussian processes and use these models to predict RSSIs at the particle's positions to obtain point estimates of the RSSI likelihood function. The particle filter's performance is assessed with real data of two test trajectories in an environment challenging for GNSS and WLAN fingerprinting localization. Outcomes of an extended Kalman filter using pseudoranges and a WLAN position as observation is included as benchmark. The proposed algorithm achieves accurate and robust seamless localization with a median accuracy of five meters.

1. Introduction

The global navigation satellite system (GNSS) is present in almost all domains of modern life and it provides the base for many applications and services ranging from transportation and logistics, via surveying and mapping, to an uncountable number of leisure activities. With the development of the Internet of Things, the amount of location-aware services and the demanded accuracy will increase further. In densely constructed urban areas and indoors, GNSS has deficits. Shadowing and multipath phenomenon, which are often the largest error sources in these situations [1], inhibit robust, ubiquitous positioning. GNSS alone is unable to satisfy many location based services.

A common strategy to achieve robust ubiquitous positioning is to support GNSS with additional sensor information. Different technologies are available and many of them have been used in combination with GNSS. Positioning systems based upon wireless local area networks (WLAN) are emerging from the large amount of options [2, 3]. This is due

to the global dissemination of WLAN infrastructure (limiting the costs of its exploitation) and WLAN enabled devices and the trade-off between range and potential positioning accuracy.

The approaches to using WLANs for localization are diverse. In this work, we resort to WLAN location fingerprinting for the reasons set forth as follows. WLAN location fingerprinting is based on the recognition of pre-recorded signal strength (RSSI) readings. These RSSIs are ideally unique at different locations. This ideal is usually not achieved. Nonetheless, a heterogeneous signal strength distribution, achieved by diverse signal attenuations, facilitates the recognition of signal strength patterns. That applies in environments with many obstacles and thereby improves the localization accuracy. Hence, this method works best indoors and in urban areas. In contrast, GNSS is based upon the signal propagation delays between the satellites and the receiver, so called pseudoranges. Exact GNSS positioning demands the line-of-sight between receiver and satellites. Open outdoor environments meet this condition best; harsh

GNSS environments are dense urban centres and indoors. The operation principles of both localization technologies differ completely. They are well suited to complement each other, thus motivating the integration of pseudoranges and received signal strengths.

Several authors studied hybrid positioning systems based on GNSS and WLAN. The approaches differ in the level of integration and in the kind of information used from each of the systems. In a survey of these approaches, we found four categories in which the fusion of information of GNSS and WLAN can be grouped [4]: (1) WLAN assisted GNSS, (2) switching between position estimates of GNSS and of a WLAN based positioning system, (3) weighting the position estimates of GNSS and of a WLAN based system, and (4) deep integration of GNSS pseudoranges with different features of a WLAN based positioning system. We refer to [4] for the implications of the different integration schemes on the overall system and continue here with a brief overview of the works that study the fusion of GNSS and WLAN raw data, as they are the most relevant. They commonly rely on recursive Bayesian estimation methods, such as Kalman filter, particle filter, and their variants. Furthermore, all of them employ GNSS pseudoranges, but the features used from WLAN differ.

The authors of [5–9] deduce ranges from WLAN signals and combine them with pseudoranges. The WLAN ranges are obtained either from signal propagation times or from WLAN power measurements. In any case, the access point positions must be known, which is an unrealistic assumption in public areas, such as city centres. Another drawback of using ranges is the requirement of a line-of-sight between transmitter and receiver. This line-of-sight condition is rare indoors and makes accurate, seamless indoor/outdoor positioning very difficult. The advantage of this approach is that the same physical quantity is fused. This means that to fuse the ranges one can use the same, well-known methods as for GNSS positioning.

We consider the fusion of pseudoranges directly with RSSIs more promising, because of their contrary error mechanisms. In [10, 11], this approach was already investigated; likelihood functions from pseudorange and RSS measurements are established in both works and then fused within a particle filter. However, as fingerprints are spatially discrete distributed, the accuracy depends strongly on the fingerprint density. This is a clear disadvantage for medium- to large-area localization applications, where the laborious construction of fingerprint radio maps presents the biggest drawback and challenge.

This study extends the previous works and provides a solution that deals with the discrete fingerprints. We present a particle filter that integrates pseudoranges and RSSIs based on their likelihood functions, where a particle approximation of the signal strength likelihood function is obtained from a Gaussian process regression model. This method provides an automatic, well balanced weighting of the two sensor information sources and achieves accurate, robust, and smooth seamless positioning. We compare the algorithm with an extended Kalman filter that feeds WLAN position estimates as observation into a regular GPS extended Kalman filter.

2. Interpolating WLAN Signal Strength

To perform seamless location estimation in this study, we consider the problem of fusing WLAN RSSI with GNSS pseudoranges on a continuous state space. Fingerprinting techniques are based on spatial samples to construct the required radio map. Interpolating RSSI provides two principle advantages. First, it reduces the labor intensive construction of radio maps, because less fingerprints are needed to obtain a fingerprint database that is equally exact compared to without interpolation. Second and more importantly here, it provides the continuous model for the fingerprint database, facilitating the fusion of RSSI and pseudoranges.

This section explains interpolation of RSSI radio maps through Gaussian process regression.

2.1. Preliminaries. Gaussian processes are stochastic processes, basically a generalization of multivariate Gaussian distributions to infinite dimension. Every point of the infinite input domain has a Gaussian random variable associated. The defining property is that any finite collection of those random variables has a joint multivariate Gaussian distribution. Gaussian processes are completely characterized by a mean function $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and a covariance function $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \quad (1)$$

where $\mathbf{x}$ is a vector from the index set.

2.2. Gaussian Process RSSI Model. Consider a nonlinear function $s_n = h(\mathbf{p}_n) + \varepsilon_n$ describing the relationship between positions in space $\mathbf{p}_n$ and the corresponding RSSIs, s_n . The term $h(\cdot)$ is the latent function we seek in order to predict/interpolate RSSIs and ε_n represents i.i.d noise. In practice, one deals with a finite collection of input points. Constructing a fingerprint radio map with M fingerprints corresponds to drawing a set of noisy RSSI samples $\mathbf{s} \equiv \{s_n\}_{n=1}^M$ at known positions $P \equiv \{\mathbf{p}_n\}_{n=1}^M$ from that unknown function. In the context of machine learning, this data is called *training data*.

Gaussian process modelling can be seen from a Bayesian point of view. The basic characteristics of the latent function is described by a Gaussian process a priori distribution and the function that we want to infer is modelled by a Gaussian process a posteriori distribution. To model the latent function appropriately, the mean and covariance function of the prior distribution have to be chosen and their parameters, so called hyperparameters, need to be found. (The hyperparameters can be learned from the training data, which is described in Section 2.3.2.) The likelihood function of the Gaussian process modelling reflects how likely are the RSSI samples in the light of the model.

The observed RSSIs are spatially correlated; neighbouring RSSIs are usually more strongly correlated than RSSIs that are distant. This relationship between the RSSIs is modelled by a covariance function. Covariance functions are specified by kernels. For two different input positions in space, $\mathbf{p}_p$

and $\mathbf{p}_q$, these functions are one at zero input distance, $\|\mathbf{p}_p - \mathbf{p}_q\|$, and decay with increasing distance. In [12], it has been shown that a constant mean function and a Matérn covariance function fit RSSI data very well. The constant mean function constitutes a single hyperparameter and is simply $[m(\mathbf{p}_p), m(\mathbf{p}_q)]^T = c$. As in [12], we use the Matérn kernel (with $\nu = 3/2$ [13]). To account for the i.i.d observation noise, the kernel is enhanced with a noise term:

$$\begin{aligned} & \text{cov}(h(\mathbf{p}_p) + \varepsilon_p, h(\mathbf{p}_q) + \varepsilon_q) \\ &= \sigma_f^2 \left(1 + \frac{\sqrt{3}}{\ell} \|\mathbf{p}_p - \mathbf{p}_q\| \right) \exp \left(-\frac{\sqrt{3}}{\ell} \|\mathbf{p}_p - \mathbf{p}_q\| \right) \quad (2) \\ &+ \sigma_\varepsilon^2 \delta(\mathbf{p}_p, \mathbf{p}_q). \end{aligned}$$

This covariance function has three hyperparameters: σ_f^2 is the signal variance, basically scaling the RSSIs; ℓ is the length scale, determining the variability of the underlying process; and σ_ε^2 is the noise variance, specifying the power of the noise. The constant mean, c , specifies the value to which the Gaussian process converges if training data are absent.

2.3. Gaussian Process Regression. The Gaussian process that describes a finite collection of observed, noisy RSSI samples is a multivariate Gaussian distribution $\mathbf{s} \sim \mathcal{N}(\mu(\mathbf{s}), \text{cov}(\mathbf{s}))$, with the mean vector $\mu(\mathbf{s}) = \mathbf{m}(P)$ and the covariance matrix $\text{cov}(\mathbf{s}) = K(P, P) + \sigma_\varepsilon^2 I$. Here, $K(P, P)$ is a matrix that contains the covariances, determined by the Matérn kernel, for all pairs of the position matrix P .

The described relationship between already observed RSSIs extends as well to not yet seen RSSIs, that is, to values we wish to predict. We denote those RSSIs with $\mathbf{s}^*$ and the corresponding positions with P^* and call the collection of these two sets *test data*. Both sets of RSSIs, the training outputs and the test outputs, originate from the same physical phenomena and are thus jointly Gaussian distributed:

$$\begin{aligned} \begin{bmatrix} \mathbf{s} \\ \mathbf{s}^* \end{bmatrix} &\sim \mathcal{N} \left(\begin{bmatrix} \mathbf{m}(P) \\ \mathbf{m}(P^*) \end{bmatrix}, \right. \\ &\left. \begin{bmatrix} K(P, P) + \sigma_\varepsilon^2 I & K(P, P^*) \\ K(P^*, P) & K(P^*, P^*) + \sigma_\varepsilon^2 I \end{bmatrix} \right). \quad (3) \end{aligned}$$

The covariance matrix $K(P, P^*) = K(P^*, P)^T$ contains the covariances evaluated at all pairs of training and test positions and $K(P^*, P^*)$ at all pairs of test positions, respectively.

By conditioning the joint Gaussian distribution (3) on the observations and inputs, the Gaussian process posterior distribution is obtained:

$$\mathbf{s}^* | P, \mathbf{s}, P^* \sim \mathcal{N}(\mu(\mathbf{s}^*), \text{cov}(\mathbf{s}^*)). \quad (4)$$

Its mean and covariance functions are given with

$$\begin{aligned} \mu(\mathbf{s}^*) &\equiv \mathbb{E}[\mathbf{s}^* | P, \mathbf{s}, P^*] \\ &= \mathbf{m}(P^*) \end{aligned} \quad (5)$$

$$\begin{aligned} &+ K(P^*, P) [K(P, P) + \sigma_\varepsilon^2 I]^{-1} (\mathbf{s} - \mathbf{m}(P)), \\ \text{cov}(\mathbf{s}^*) &= [K(P^*, P^*) + \sigma_\varepsilon^2 I] \\ &- K(P^*, P) [K(P, P) + \sigma_\varepsilon^2 I]^{-1} K(P, P^*). \end{aligned} \quad (6)$$

Equation (5) enables the prediction of RSSIs at arbitrary positions P^* and (6) enables computing the covariance at these positions. These covariances provide a useful measure of the uncertainty of the predictions.

2.3.1. Nonzero Mean Function. The term $\mathbf{m}(\cdot)$ is a deterministic mean function. Regression with a fixed mean function is simple. The fixed mean is subtracted from the observations, reverting the Gaussian process to a zero mean process. Then the standard Gaussian process regression procedure is applied, after which the mean function is added again. By replacing the fixed mean function with a set of basis functions, the mean function can be parametrized. Its hyperparameters, the parameters of the basis functions, can then be inferred from the data [13].

2.3.2. Finding the Hyperparameters. A final point we need to mention is how to obtain the hyperparameter of the mean and covariance function. The likelihood function of Gaussian process regression, $p(\mathbf{s} | P, \theta)$, is the key to that.

For notational convenience, we collect the hyperparameters in a vector $\theta = \{c, \sigma_f, \ell, \sigma_\varepsilon\}$. The posterior distribution's dependence on the hyperparameter is expressed by

$$p(\mathbf{s}^* | P^*, \mathbf{s}, P) = \int p(\mathbf{s}^* | P^*, \mathbf{s}, P, \theta) p(\theta | \mathbf{s}, P) d\theta, \quad (7)$$

an intractable integral. Nonetheless, it can be approximated by using the most probable values of the hyperparameters $p(\mathbf{s}^* | P^*, \mathbf{s}, P) \approx p(\mathbf{s}^* | P^*, \mathbf{s}, P, \theta_{\text{mp}})$ [14]. The most probable hyperparameters θ_{mp} can be determined based on the posterior probability of θ , $p(\theta | \mathbf{s}, P)$. Maximizing

$$\hat{\theta} = \arg \max_{\theta} \frac{p(\mathbf{s} | P, \theta) p(\theta)}{p(\mathbf{s} | P)} \quad (8)$$

allows inferring the hyperparameters. This fraction reduces to the marginal likelihood function, as the denominator is independent of θ and $p(\theta)$ is modelled uniformly (usually no prior knowledge about the hyperparameters is available). In practice, the inference is done by minimizing the negative log-likelihood:

$$\hat{\theta} = \arg \min_{\theta} (-\ln p(\mathbf{s} | P, \theta)). \quad (9)$$

Assuming a normally distributed noise process and prior distribution, an analytic expression for the marginal likelihood can be derived [13]:

$$\ln p(\mathbf{s} | P, \theta) = -\frac{N}{2} \ln 2\pi - \frac{1}{2} \ln |K + \sigma_e^2 I| - \frac{1}{2} (\mathbf{s}^T (K + \sigma_e^2 I)^{-1} \mathbf{s}). \quad (10)$$

The optimal hyperparameters are found by computing the partial derivatives of this function with respect to the hyperparameters and then applying a gradient search algorithm to find them.

3. Sequential Bayesian Estimation

The localization problem can be optimally described by the sequential Bayesian filter, a recursive framework that also enables fusing the sensory data. It propagates the location estimate over time based on the previous estimate of the mobile terminal's position and arriving observations.

It consists of two steps that are recursively executed. The process update predicts the current location based on the previous one and a model of the mobile terminal's motion. The process update is expressed by the Chapman-Kolmogorov equation [15]:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) = \int p(\mathbf{x}_k | \mathbf{x}_{k-1}) p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1}) d\mathbf{x}_{k-1}, \quad (11)$$

where $\mathbf{x}_k$ is the state vector at discrete time k . The time index $1 : k$ denotes a sequence of variables, for example, all observations, $\mathbf{z}_{1:k} \equiv \{z_i, i = 1, \dots, k\}$, obtained up to time k . The motion of the mobile terminal is modelled by the state transition probability density $p(\mathbf{x}_k | \mathbf{x}_{k-1})$. The term $p(\mathbf{x}_{k-1} | \mathbf{z}_{1:k-1})$ is the prior probability density. It expresses information about the mobile terminal available before the current estimation step and represents the previous state of the object or some initial state. As soon as observations are available, the predicted state can be corrected. This is done in the measurement update step:

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) = \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1})}{\int p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{z}_{1:k-1}) d\mathbf{x}_k}, \quad (12)$$

where $p(\mathbf{z}_k | \mathbf{x}_k)$ is the likelihood function [15]. It encodes the information of the observations and is used to correct the prediction to yield the posterior density $p(\mathbf{x}_k | \mathbf{z}_{1:k})$. This recursion estimates the location of the mobile terminal optimally; however, these integrals do usually not have an analytic solution and must be approximated.

3.1. Particle Filter. One of the most versatile approximations to the recursive Bayesian estimation problem is particle filters. Particle filters rely on sampling from functions to approximate them. The versatility stems from their ability to approximate arbitrary functions. A detailed description of the generalized particle filter can be found in [15], that we follow in this subsection.

Complex integrals, as the ones that describe the sequential Bayesian estimator, can be approximated numerically if we can sample the involved functions. If the samples cover the function's domain well, sums replace complex integrals while the simplified operation on the samples provides a sufficiently accurate approximation. We denote the samples by $\{x^{(i)}, i = 1, \dots, N\}$.

Nevertheless, it is often not possible to sample a function, as it is unknown or too complex. This issue is circumvented by sampling a similar function, $q(x)$, which is proportional to the target function $p(x^{(i)}) = \tilde{\omega}^{(i)} q(x^{(i)})$. The term $q(x^{(i)})$ is the importance density. The introduced weights $\tilde{\omega}^{(i)}$ compensate the difference between the importance density and the target density. These weights need to be normalized to sum to unity, so that the functions are valid probability densities: $\omega^{(i)} = \tilde{\omega}^{(i)} / \sum_{j=1}^N \tilde{\omega}^{(j)}$. The idea of approximating arbitrary functions by a set of weighted particles $\{x^{(i)}, \omega^{(i)}\}_{i=1}^N$ is known as importance sampling. Consider the samples $\{x^{(i)}, i = 1, \dots, N\} \sim q(x^{(i)})$. Thus, we can approximate a density $p(x)$ by samples:

$$p(x) \approx \sum_{i=1}^N \omega^{(i)} \delta(x - x^{(i)}). \quad (13)$$

In the context of recursive state estimation, $\mathbf{x}^{(i)}$ is the state of the i th particle and $\omega^{(i)}$ is its associated weight. The target density is the posterior density $p(\mathbf{x}_k | \mathbf{z}_{1:k})$. To this point, it was left open how the weights are obtained. They are first defined in general by

$$\omega_k^{(i)} \propto \frac{p(\mathbf{x}_{0:k}^{(i)} | \mathbf{z}_{1:k})}{q(\mathbf{x}_{0:k}^{(i)} | \mathbf{z}_{1:k})}. \quad (14)$$

The derivation of the weights is based on a factorization of the importance density. A factorization that allows augmenting the importance density of the previous time step with a factor that represents the current state, so that particles that are distributed according to the importance density of the previous state can be updated with particles that approximate the current state. This leads to a recursive weight update which is given by

$$\omega_k^{(i)} \propto \omega_{k-1}^{(i)} \frac{p(\mathbf{z}_k | \mathbf{x}_k^{(i)}) p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})}{q(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k)}. \quad (15)$$

Equation (15) allows approximating the posterior density by particles and their weights.

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) \approx \sum_{i=1}^N \omega_k^{(i)} \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}). \quad (16)$$

The weights must be normalized after each iteration.

The described procedure constitutes the problem of particle degeneration; that is, after each of the iterations, only a few particles gain weights while, for the majority of particles, the weight is reduced, until just one particle concentrates almost

the complete weight and the weight of the remaining particles is negligible.

The choice of the importance density may slow down this effect, but usually an additional resampling (with replacement) step is introduced to overcome it. During the resampling step, particles are multiplied according to their weights, so that particles that possess large weights are selected several times and particles with very small weights are selected once or possibly vanish.

As the resampling introduces further problems (principally reduces the diversity, because many particles are repeated), we resample when the degeneracy of the algorithm becomes severe. This can be assessed by the effective sample size $N_{\text{eff}} = 1 / \sum_{i=1}^N (\omega_k^{(i)})^2$. We resample if the effective sample size exceeds $2/3$ of the numbers of particles.

In this work, we choose the importance density to be the state transition density $q(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)}, \mathbf{z}_k) = p(\mathbf{x}_k^{(i)} | \mathbf{x}_{k-1}^{(i)})$. This simplifies (15) further; it reduces the weight update to be a product of the previous weights and the likelihood function:

$$\omega_k^{(i)} \propto \omega_{k-1}^{(i)} p(\mathbf{z}_k | \mathbf{x}_k^{(i)}). \quad (17)$$

4. Fusing GNSS Pseudoranges and WLAN Received Signal Strength

This section starts with an explanation of the used models in Section 4.1 and describes their integration into the Bayesian filter framework in Section 4.2. The extended Kalman filter that is used as benchmark is detailed at the end of this section.

4.1. Models. The state space for the self-localization problem is defined in a local level coordinate frame, where a position in space is denoted by $\mathbf{p} = (x, y, z)^T$. The corresponding velocity is its time derivative $\dot{\mathbf{p}} = (\dot{x}, \dot{y}, \dot{z})^T$. We denote the state vector at time k by $\mathbf{x}_k = (\mathbf{p}_k, \dot{\mathbf{p}}_k)^T$.

4.1.1. Process Model. To model the motion of the mobile terminal, we have chosen a random walk velocity process as in [16]. It stems from a Langevin process, a stochastic differential equation, that describes the velocity of a Brownian particle. For motion in a single dimension, this process reads

$$\begin{aligned} \dot{x}_k &= a_x \dot{x}_{k-1} + b_x v_{x_k}, \\ x_k &= x_{k-1} + \Delta T \dot{x}_k \end{aligned} \quad (18)$$

with

$$\begin{aligned} a_x &= \exp(-\beta_x \Delta T), \\ b_x &= \bar{v}_x \sqrt{1 - a_x^2}. \end{aligned} \quad (19)$$

It is driven by a white and normally distributed excitation process $v_{x_k} \sim \mathcal{N}(0, 1)$, which allows the uncertainty of the velocity to grow during prediction and also enhances the diversity of the particles. Parameter β_x is the process' rate constant. It determines the velocity increment of the mobile terminal during a discretization time step ΔT , where steady-state root-mean-square velocity, $\bar{v}_x$, confines the velocity of

the mobile terminal. For the experiments described above, we determined these parameters empirically to $\beta_{x,y} = 10 \text{ s}^{-1}$, $\beta_z = 100 \text{ s}^{-1}$ and $\bar{v}_{x,y} = 1.5 \text{ m s}^{-1}$, and $\bar{v}_z = 0.5 \text{ m s}^{-1}$.

This model results in a linear process model:

$$\mathbf{x}_k = F_{pv} \mathbf{x}_{k-1} + G_{pv} \mathbf{v}_{k-1}, \quad (20)$$

with the state transition and noise coupling matrix

$$F_{pv} = \begin{pmatrix} 1 & 0 & 0 & a_x \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 & a_y \Delta T & 0 \\ 0 & 0 & 1 & 0 & 0 & a_z \Delta T \\ & & & a_x & 0 & 0 \\ \mathbf{0}_{3 \times 3} & & & 0 & a_y & 0 \\ & & & 0 & 0 & a_z \end{pmatrix}, \quad (21)$$

$$G_{pv} = \begin{pmatrix} b_x & 0 & 0 \\ 0 & b_y & 0 \\ 0 & 0 & b_z \\ b_x \Delta T & 0 & 0 \\ 0 & b_y \Delta T & 0 \\ 0 & 0 & b_z \Delta T \end{pmatrix}.$$

The noise covariance matrix is simply $V \equiv \mathbb{E}[\mathbf{v}_k, \mathbf{v}_k] = I$; thus, the state transition probability density is Gaussian with the corresponding mean and covariance matrix:

$$\mathbf{x}_k | \mathbf{x}_{k-1} \sim \mathcal{N}(F_{pv} \mathbf{x}_{k-1}, G_{pv} V G_{pv}^T). \quad (22)$$

This process model is rather unspecific and therefore very general. Matrix F_{pv} implies basically a static mobile terminal; no specific motion or direction is presumed. Instead it increases the variance over time, thus broadening the region of the potential location. The choice of parameters increases the variance in east and north equally but restricts it in z-direction. The reason for that setting is attributed to the lack of an useful geographic height in the radio map. As the z-coordinate cannot be corrected in GPS denied areas, we basically slow down the motion in z-direction considerably.

4.1.2. GNSS Pseudorange Likelihood Function. A pseudorange describes the distance from the satellite to the receiver. This consists of the geometric distance, the difference between the satellite clocks and the receiver clock, and an ionospheric and tropospheric correction term. Most of these terms can be corrected using information transmitted by the satellites. However, errors occurring in the user segment remain. This is the receiver clock offset, Δt_k , which can usually be estimated. Other considerable errors are attributed to multipath propagation and shadowing effects. We collect these in $w_{\eta,k}$, together with errors from the ephemerides

prediction, relativistic effects, residuals from the correction terms, and noise. The pseudorange of the j th satellite becomes

$$\rho_k^j = \|\mathbf{p}_{s,k}^j - \mathbf{p}_k\| + c\Delta t_k + w_{\eta,k}^j, \quad j = 1, \dots, J. \quad (23)$$

The error term $w_{\eta,k}$ is commonly modelled as a normally distributed, zero mean random variable $w_{\eta,k} \sim \mathcal{N}(0, \sigma_{w_{\eta,k}}^2)$. Assuming furthermore that the pseudoranges from different satellites are independent, the joint pseudorange likelihood function for the collection of pseudoranges, $\boldsymbol{\rho}_k = \{\rho_k^j\}_{j=1}^J$, can be written as

$$p(\boldsymbol{\rho}_k | \mathbf{x}_k = \mathbf{x}_q) = \prod_{j=1}^J \frac{1}{\sqrt{2\pi}\sigma_{w_{\eta,k}}^j} \exp\left(-\frac{1}{2} \frac{(\bar{\rho}_k^j - \rho_k^j)^2}{(\sigma_{w_{\eta,k}}^j)^2}\right). \quad (24)$$

It expresses the likelihood that the observed pseudoranges were measured at $\mathbf{p}_q = C\mathbf{x}_q \in \mathbb{R}^3$, where $C = (I_3 \mid \mathbf{0}_3)$. The standard deviation $\sigma_{w_{\eta,k}}^j$ is composed by various terms that are set or estimated by RTKLIB [17]; among these terms are a deviation estimate based on the user accuracy index and fix values set by RTKLIB accounting for the errors in estimating delays attributed to the troposphere, ionosphere, and code bias.

Notice that the receiver clock offset is not part of the state vector and it is not estimated by the particle filter. As in [11], the least squares method is used iteratively to estimate the receiver clock offset.

4.1.3. WLAN RSSI Likelihood Function. A likelihood function for the RSSI observations, s_k , is established as follows. As we work with time averaged RSSI, to compensate the variations and to get closer to the assumption of normally distributed observations, we denote the arithmetic mean of RSSIs by $\bar{s}_k$. To evaluate the likelihood of an observation at different positions, a model predicting the observation, $\hat{s}_k$, is required. However, the lack of an accurate model that relates RSSI and space is the reason to resort to the empirical method of fingerprinting. As RSSI measurements only exist at fingerprint locations, one has to approximate the RSSI measurement at the predicted position of the mobile terminal.

These RSSI estimates are obtained through Gaussian process regression; recall Section 2.3. Assume we have $l = 1, \dots, L$ access points and that we receive their packets during survey and localization phase. The Gaussian process model for each access point can be trained off-line, before the actual positioning phase. Let $\bar{s}_k^l$ be the average of RSSIs received from the l th access point. Once the Gaussian process models are constructed, one uses (5) and (6) to predict a vector of RSSIs, $\mu(\mathbf{s}_k^{*,l})$, and the corresponding covariance matrix, $\text{cov}(\mathbf{s}_k^{*,l})$, at the positions of interest P_k^* .

Finally, to compute the likelihood that $S_k = \{\bar{s}_k^l\}_{l=1}^L$ were observed at the position $\mathbf{p}_q^* \in P^*$, we establish a Gaussian likelihood function:

$$p(S_k | \mathbf{x}_k = \mathbf{x}_q) = \prod_{l=1}^L \frac{1}{\sqrt{2\pi}\sigma_{s^*}^l} \exp\left(-\frac{1}{2} \frac{(\bar{s}_k^l - \mu_{s^*}^l)^2}{(\sigma_{s^*}^l)^2}\right), \quad (25)$$

where $\mu_{s^*}^l$ is the interpolated RSSI and $\sigma_{s^*}^l$ the corresponding standard deviation at the test input $\mathbf{p}_k^*$. This likelihood function is based on the assumption that the RSSI observations are independent. Notice, the radio map contains only two-dimensional location information; therefore $\mathbf{p}_q = D\mathbf{x}_q \in \mathbb{R}^2$ and the Boolean matrix is $D = (I_2 \mid \mathbf{0}_{2 \times 4})$.

Compared with the pseudorange likelihood function, the predicted RSSI and variance are independent of time, because they are inferred from the (static) radio map. The same test position yields the same prediction. In contrast, the standard deviation in the pseudorange likelihood function is independent of the location of the mobile platform; it is only determined by the signal reception conditions. The choice of the RSSI variance arises from measurement setup. For many access points, only a single packet was captured; hence, a reasonable deviation measure could not be estimated. An alternative would be to exclude the observations from these access points and then use only the sample deviation measure from the remaining RSSI.

4.2. Particle Filter Tight Fusing of Pseudoranges and Signal Strength. This section combines the ideas previously described; it details the fusion of GPS pseudoranges and WLAN RSSI within the particle filter. Briefly, the key probability density functions of the sequential Bayesian estimator are approximated with particles and their weights and then plugged into the particle filter of Section 3.1.

4.2.1. Process Update. The probability density function that models the motion is already stated in (22). One could sample that density, compute the particle approximation to (11), and predict the mobile terminal's new location. Since an analytic process model is available, it is easier to propagate the particles that approximate the prior density, $p(\mathbf{x}_{k-1}^{(i)} | \mathbf{z}_{1:k-1})$, through the process model (20):

$$\mathbf{x}_k^{(i)} | \mathbf{z}_{1:k-1} = F_{pv}\mathbf{x}_{k-1}^{(i)} + G_{pv}\mathbf{v}_{k-1}^{(i)}, \quad (26)$$

where $\{\mathbf{v}_{k-1}^{(i)}\}_{i=1}^N$ are samples from the noise model.

When the filter is initialized, the prior density is usually not known. A common choice for $p(\mathbf{x}_0) \equiv p(\mathbf{x}_0 | \mathbf{z}_{-1})$ is a uniform density over a plausible area.

4.2.2. Measurement Update. The measurement update is formally defined by (12). The particle filter computes the posterior density through (16) and (17). At this step, a particle approximation of the predictive density is already available $p(\mathbf{x}_k^{(i)} | \mathbf{z}_{1:k-1})$, as are the weights from the previous time step.

Initially and after resampling, the weights are all equal and sum up to unity. What is left is to update the weights via the likelihood function.

A particle approximation of the pseudorange likelihood is simply obtained by computing the likelihood for each particle:

$$p(\rho_k | \mathbf{x}_k = \mathbf{x}_{k_n}^{(i)}) = \prod_{j=1}^J \frac{1}{\sqrt{2\pi}\sigma_{w_{\eta,k}}^j} \exp\left(-\frac{1}{2} \frac{(\bar{\rho}_k^j - \rho_k^{j(i)})^2}{(\sigma_{w_{\eta,k}}^j)^2}\right), \quad (27)$$

where $\rho_k^{j(i)}$ is a prediction of the j th pseudorange for a particle position according to the pseudorange model.

We proceed similarly for the RSSI likelihood function and compute the likelihood for each particle. This involves predicting a RSSI and variance value at each position of a particle; the RSSI likelihood function for the test positions $P_k^* = \{D\mathbf{x}_{k_n}^{(i)}\}_{i=1}^N$ becomes

$$p(S_k | \mathbf{x}_k = \mathbf{x}_{k_n}^{(i)}) = \prod_{l=1}^L \frac{1}{\sqrt{2\pi}\sigma_{s^*,k}^{l(i)}} \exp\left(-\frac{1}{2} \frac{(\bar{s}_k^l - \mu_{s^*,k}^{l(i)})^2}{(\sigma_{s^*,k}^{l(i)})^2}\right). \quad (28)$$

The term $\mu_{s^*,k}^{l(i)}$ denotes the RSSI predictions at the particle's positions and $\sigma_{s^*,k}^{l(i)}$ is the standard deviation at these positions, respectively.

Pseudoranges and RSSIs are clearly independent of each other, thus fusing RSSIs and pseudoranges equals multiplying their likelihood functions. An approximation of the posterior density is given by

$$p(\mathbf{x}_k | \mathbf{z}_{1:k}) \approx \sum_{i=1}^N \omega_{k-1}^{(i)} p(S_k | \mathbf{x}_k^{(i)}) p(\rho_k | \mathbf{x}_k^{(i)}) \delta(\mathbf{x}_k - \mathbf{x}_k^{(i)}). \quad (29)$$

Likelihood functions provide information on the parameters only up to a multiplicative constant. To yield a valid probability density, the updated weights have to be normalized so that they sum up to unity.

In situations when no observation from one of the sensors was obtained, the likelihood function cannot be computed and is therefore ignored. If no measurement at all is available, the algorithm proceeds directly to the process update.

The influence of the measurements, either a pseudorange or a RSSI average, on the particles is controlled by these likelihood functions. On the one hand, the variances affect the particles weights. In the case of pseudoranges, it increases if the GPS observations are believed to be inexact or noisy. The variance of the RSSI likelihood function increases if the mobile terminal is in an area with few fingerprints, because, in these areas, information about RSSI is little. Large variances lead to small weights and vice versa. On the other hand, the position of a particle also affects its weight, because a

particle far from the location indicated by an observation causes a large difference in the exponential of the functions, which decreases the weight. Due to these mechanisms, the proposed particle filter balances automatically between the two positioning systems.

4.2.3. Position Estimate. The final position estimate results from the minimization of the mean squared error, $\hat{\mathbf{x}} = \arg \min_{\hat{\mathbf{x}}} \mathbb{E}[(\hat{\mathbf{x}} - \mathbf{x})^2 | \mathbf{z}]$ [18], and is known as the expected a posteriori estimate:

$$\hat{\mathbf{x}}_k = \mathbb{E}[\mathbf{x}_k | \mathbf{z}_k] \equiv \int \mathbf{x}_k p(\mathbf{x}_k | \mathbf{z}_k) d\mathbf{x}_k, \quad (30)$$

for the particle filter that becomes $\hat{\mathbf{x}}_k = \sum_{i=1}^N \omega_k^{(i)} \mathbf{x}_k^{(i)}$.

We compute also the standard deviation of the particles as quality measure of the point estimate:

$$\sigma_p = \sqrt{\mathbb{E}[(\hat{\mathbf{x}}_k - \mathbf{x}_k^{(i)})^2]} = \sqrt{\frac{\sum_{i=1}^N \omega_k^{(i)} (\hat{\mathbf{x}}_k - \mathbf{x}_k^{(i)})^2}{(N' - 1) \sum_{i=1}^N (\omega_k^{(i)}) / N'}}, \quad (31)$$

where N' denotes the nonzero weights [19].

4.3. Benchmark Extended Kalman Filter. To put the results of the proposed particle filter into context, we present them along with results obtained from an extended Kalman filter. This filter is based upon the process and measurement update stated in (11) and (12), but instead of propagating and correcting the probability densities it assumes the noises to be zero mean and normally distributed and therefore it propagates only the mean, $\hat{\mathbf{x}}$, and the error covariance matrix, P , in time. Furthermore, we extend the state vector by the receiver clock offset, Δt_k , and its time derivative $\Delta \dot{t}_k$: $\mathbf{x}_k = (\mathbf{p}_k, \dot{\mathbf{p}}_k, \Delta t_k, \Delta \dot{t}_k)^T$.

We present first the process and measurement models and then the update and the correction step of the extended Kalman filter.

4.3.1. Process Model. The motion of the mobile terminal is described by the linear model (20). The process model of the receiver clock is according to [20] a two-state random process model

$$\begin{pmatrix} \Delta t_k \\ \Delta \dot{t}_k \end{pmatrix} = F_t \begin{pmatrix} \Delta t_{k-1} \\ \Delta \dot{t}_{k-1} \end{pmatrix} + G_t \begin{pmatrix} v_{1_{k-1}} \\ v_{2_{k-1}} \end{pmatrix}, \quad (32)$$

where the state transition matrix reads

$$F_t = \begin{pmatrix} 1 & \Delta T \\ 0 & 0 \end{pmatrix}. \quad (33)$$

The noise coupling matrix, G_t , can be obtained from the covariance matrix of the clock offset noise by $G_t = W\sqrt{D}$, where W denotes the eigenvectors and D the eigenvalues of Q_t ; $\sqrt{\cdot}$ stands for the square root of the matrix' elements. The elements of the noise covariance matrix itself are determined

by the characteristics of the clock, given through the Allan variance coefficients h_0 , h_{-1} , and h_{-2} :

$$\begin{aligned} q_{11} &= \frac{h_0}{2}\Delta T + 2h_{-1}\Delta T^2 + \frac{2}{3}\pi^2 h_{-2}\Delta T^3, \\ q_{12} &= h_{-1}\Delta T + \pi^2 h_{-2}\Delta^2, \\ q_{21} &= q_{12}, \\ q_{22} &= \frac{h_0}{2\Delta T} + 4h_{-1} + \frac{8}{3}\pi^2 h_{-2}\Delta, \end{aligned} \quad (34)$$

which can be found in [20] (We assume the GPS receiver clock to have a temperature compensated crystal clock.).

Combining the state transition and the noise coupling matrices of the motion and the receiver clock model

$$\begin{aligned} F &= \left(\begin{array}{c|c} F_{pv} & \mathbf{0}_{6 \times 2} \\ \hline \mathbf{0}_{2 \times 6} & F_t \end{array} \right), \\ G &= \left(\begin{array}{c|c} G_{pv} & \\ \hline G_t & \mathbf{0}_{2 \times 1} \end{array} \right), \end{aligned} \quad (35)$$

we can write the complete process model:

$$\mathbf{x}_k = F\mathbf{x}_{k-1} + G\mathbf{v}_{k-1}. \quad (36)$$

4.3.2. Measurement Model. Measurements to the extended Kalman filter are the pseudoranges and, if available, a position from WLAN. Thus, the measurement vector is $\mathbf{z}_k = (\rho_k^1, \dots, \rho_k^J, \mathbf{p}_{\text{WLAN}})^T$. The measurement model is the common GPS measurement model for an eight-state extended Kalman filter; see [20]. Recalling the pseudorange model (23), we will abbreviate it here with $h(\cdot)$. Its Jacobian matrix reads

$$H_p = \begin{pmatrix} h_x^1 & h_y^1 & h_z^1 & 0 & 0 & 0 & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_x^J & h_y^J & h_z^J & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, \quad (37)$$

where h_x^j stands for the partial derivative of the pseudorange measurement model (23) with respect to $\mathbf{x}$ for the j th satellite.

In the case that a position estimate from WLAN is available, the measurement matrix becomes simply

$$H = \left(\begin{array}{c|c} H_p & \\ \hline I_{3 \times 3} & \mathbf{0}_{3 \times 5} \end{array} \right). \quad (38)$$

The position estimated from WLAN is a maximum likelihood estimate obtained from (28):

$$\hat{\mathbf{x}}_{\text{WLAN}} = \arg \max_{\mathbf{x}} p(\mathbf{S}_k | \mathbf{x}_k), \quad (39)$$

where $\hat{\mathbf{p}}_{\text{WLAN}}$ corresponds to the first three elements of $\hat{\mathbf{x}}_{\text{WLAN}}$.

For completeness, we denote the general measurement model including all observed pseudoranges and a position from WLAN based system:

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{w}_k. \quad (40)$$

The noise term, $\mathbf{w}_k$ is, as in (23), zero mean and white and follows a normal distribution with covariance matrix $R_k \equiv \mathbb{E}[\mathbf{w}_k, \mathbf{w}_i]$.

4.3.3. Process Update. In the time update step of the extended Kalman filter, the mean, $\hat{\mathbf{x}}$, and the error covariance matrix, P , are propagated in time [15, 20]:

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= F\hat{\mathbf{x}}_{k-1}, \\ P_{k|k-1} &= FP_{k-1}F^T + Q, \end{aligned} \quad (41)$$

where Q denotes the process noise covariance matrix. The final noise covariance matrix results as block matrices of $Q_{pv} = G_{pv}VG_{pv}^T$ and Q_t (given by (34)):

$$Q = \left(\begin{array}{c|c} Q_{pv} & \mathbf{0}_{6 \times 2} \\ \hline \mathbf{0}_{2 \times 6} & Q_t \end{array} \right). \quad (42)$$

4.3.4. Measurement Update. The extended Kalman filter approximates (12) as follows [15, 20]:

$$\begin{aligned} \hat{\mathbf{x}}_k &= \hat{\mathbf{x}}_{k|k-1} + K(\mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1})), \\ P_{k|k-1} &= (I + K_k H_k) P_{k|k-1}. \end{aligned} \quad (43)$$

Matrix K is the Kalman gain and is given by $K = P_{k|k-1}H_k (H_k P_{k|k-1}H_k^T + R_k)^{-1}$. Here, $R_{p,k}$ is the measurement covariance matrix that contains a variance estimate estimated by RTKLIB for each pseudorange on its diagonal. These variance estimates are based on the user range accuracy index. If a WLAN position is obtained, the covariance matrix is extended by a diagonal block containing a variance for each coordinate of the WLAN position:

$$R = \left(\begin{array}{c|c} R_p & \mathbf{0} \\ \hline \mathbf{0} & R_{\text{WLAN}} \end{array} \right). \quad (44)$$

We found a standard deviation of $\sigma_{\text{WLAN}} = 0.5$ to achieve the lowest errors for both trajectories, although we expected that larger values would model the error of the WLAN maximum likelihood position estimate better [21].

5. Experimental Results

To analyze the performance of the particle filter, we conducted experiments for two trajectories. We begin this section with a description of these experiments and continue with some notes concerning the implementation of the particle filter. The outcomes of these experiments are reported subsequently in Sections 5.3 and 5.4.

5.1. Experimental Setup and Data Recording. The data used to evaluate our algorithm is from a test bed at the Universidad Autónoma de Querétaro. The test bed constitutes two relatively small buildings (about 8×40 meters) with roofed passageways through which the buildings are entered. These two buildings and a larger four-storey building in the south west surround a larger open space. In addition to the roofed passageways, several trees block the view to the sky. Figure 1 depicts the test bed and the two trajectories used to evaluate the algorithm. The first trajectory, named Trajectory-1, commences indoors without GPS observations and

FIGURE 1: Test bed and trajectories. Circles indicate starting point of the trajectories. (Notice, the trajectory actually followed the existing paths, which are illustrated by densely dotted orange lines. The markers of the trajectories are the positions at which the WLAN data was recorded; they are obtained from the measurements overlaid on the test bed map. The mismatch between the trajectories and the background image is due to errors of the overlay procedure, errors of the experimenter marking the measurement positions on the map, and issues when dealing with geographic data, such as projections and coordinate transformations.)

runs mainly indoors and along the semiopen passageways along the buildings. It enters the right building only once and ends in the open space, nearby some trees. The second trajectory is called Trajectory-2. It starts in the open space and continues on it, passes below trees, and leads to the passageway of the right building. Following that passageway, it enters the right building once and proceeds to the roofed passageway of the left building. From the middle of the left building, it runs around the small lawn area and returns on the passageway to the passageway of the right building, from where it turns sharp right towards the open space. Trajectory-2 ends further southwest on the open space, in front of a four-storey building. Since Trajectory-1 has larger indoor sections, we refer to it as indoor-like trajectory. The opposite is true for Trajectory-2; we call it outdoor-like trajectory.

This localization scenario presents a harsh environment for GNSS, as scattering, blockage and multipath propagation are very likely. The two buildings used in the indoor sections of the experiments are relatively lightweight. They are made of brick walls and the sectioning indoors is principally done by soft-partitions. As the climate favours open doors and windows, these buildings attenuate signals rather little; thus, the indoor environment is challenging for WLAN fingerprinting as well.

We recorded the RSSIs with a laptop with a consumer-grade wireless network interface card. To create the radio maps, RSSIs and MAC addresses need to be captured at certain, known locations. Therefore we modified JMapViewr, a map application providing access to OpenStreetMap data. By a click on the map, at the location that corresponds to

that of the experimenter, the position is obtained and the WLAN data capturing process is initialized: RSSIs from all available access points and the corresponding MAC addresses are captured. A second click on the user interface, after three to five seconds, issues a command to compute the arithmetic mean and the variance of the RSSIs from each access point. The capturing process finishes by storing the WLAN data and the corresponding position (obtained by from the first click on the map) in a database.

This software and procedure were also used to record the WLAN data for the test trajectories. This resulted in a stop-and-go motion, which may not be the most common motion of a pedestrian, but this facilitates recording a ground truth. Again, the WLAN data at each position of a trajectory was recorded for a few seconds. The ground truth of both paths is depicted in Figure 1.

The data sets of the trajectories also contain GPS raw data, from which pseudoranges, satellite positions, and pseudorange correction terms were obtained. This data was recorded continuously in time. The used receiver is an u-blox LEA-6T-0 GPS receiver ([22]). To parse and preprocess the data from the GPS receiver, we used RTKLIB.

Time synchronization is achieved by associating a Julian date time-stamp directly after parsing the WLAN and GPS data, respectively. After adding the time-stamps, the data was postprocessed and subsequently stored in the databases. Time synchronization is at least accurate up to half a second, enough to estimate the location of a (walking) pedestrian every second.

During data recording, the experimenter held the laptop in front of his chest, the GPS antenna was mounted on the experimenters head, whereas the WLAN antenna was connected directly to the laptop. The experimenter's body probably influenced the WLAN RSSIs and his head movements may have affected the GNSS data reception. Between the radio map survey phase and localization phase, there are more than 18 months. A degradation of the radio map is likely, nonetheless not severe, because the university's official WLAN infrastructure is relatively stable.

The radio map (see Supplementary Material in [12]) was transformed to a local coordinate frame with an origin at $(20.59^\circ, -100.415^\circ, 1800.0 \text{ m})$ and the developed filter was also implemented to compute positions in this frame.

5.2. Filter Implementation Details. Before the observation data is processed, it is synchronized with help of the assigned time-stamps.

The reception of all necessary GPS data, especially ephemerides, takes a few seconds. For that reason and the knowledge that a WLAN based location estimate is likely inexact in that area, we skip the first 35 seconds of Trajectory-2 and start the estimation with GPS observations available.

To initialize the filter we distributed the particles uniformly over the test bed and set the velocity to zero. The initial position estimate is close to the centroid of the initial particle distribution and thus relatively close to the centre of the test bed.

FIGURE 2: Position estimates of the particle filter in GPS + WLAN, GPS-only, and WLAN-only configuration and of the extended Kalman filter and the ground truth of Trajectory-1.

We repeat the estimation of a trajectory 50 times with random initialization of the particles. The results, that is RMSE, standard deviation, and the data to create the figures, are averages over these 50 repetitions.

The particle filter operates with 1000 particles. This assures a certain reliability of the filter outcomes.

5.3. Indoor-Like Trajectory: Trajectory-1. Figure 2 depicts the trajectory estimated by the particle filter (in three configurations), by the extended Kalman filter and the ground truth data. The path estimated by the particle filter follows the ground truth quite well. Visible are jumps of the position estimates. They are caused by a lack of observations. That is because RSSIs are received only every 2 s to 5 s. If, in that time interval, no GPS observations are obtained, the filter can only predict but not correct that prediction, which does not alter the position estimate. In the moment that again a reliable RSSI measurement is received, the filter corrects the prediction and the position estimate jumps to the new corrected estimate. The last section consists of an indoor part immediately followed by an outdoor part. When the trajectory enters the building (around (247, 94) m), the estimated location is already off by a few meters and also the path inside the room is not that accurately estimated. After leaving the room, the estimates drift further and the error increases. During the last section of the trajectory, the particle filter estimates follow the ground truth poorly with an offset. The extended Kalman filter shows a very poor performance. The trajectory shows large position jumps. A comparison of the extended Kalman filter estimates with that from the particle filter suggests that the extended Kalman filter takes the position estimate from WLAN not much into account; if it does, these large jumps occur. This was not expected, because the variance of the WLAN position estimate is set lower than former experiments suggested. Furthermore, the estimates of the extended Kalman filter on the roofed passageway of the

FIGURE 3: Error of the particle filter in GPS + WLAN, GPS-only, and WLAN-only configuration and of the extended Kalman filter.

right building are pulled into the open space. This indicates that the extended Kalman filter weights the GPS information much more than the particle filter does, even though the GPS data are apparently of low quality. Only towards the end of the path, when outdoors on the open space, the extended Kalman filter estimates get better, even better than those of the particle filter.

The availability of sensory data confirms the described behaviour. Information about the availability can be deduced from Figure 3. This figure depicts the magnitude error over time for the three particle filter configurations and for the extended Kalman filter. The error of the WLAN stand-alone solution indicates that WLAN data is available almost all the time. The error is low, except after the initialization, when no measurements are received for some seconds, and during the outdoor section, where WLAN location fingerprinting is known to perform poorly. The GPS stand-alone solution yields low errors during the first minutes or so, although no GPS data is received. This is due to the initial position that is relatively close to the true start position. The similarity of errors of the particle filter hybrid solution and the WLAN-only solution implies that no GPS data was received for the largest time of the trajectory, or it at least implies that GPS information was not fused into the hybrid solution.

Taking into consideration that WLAN data was available all the time and GPS-only towards the end, the accuracy of the hybrid solution can be explained in more detail: up to minute 24.5, the estimates are governed by the RSSI observations. Therefore, the errors are small where fingerprints can be distinguished well, indoors and most of the time along the passageways. (The reader is referred to [21] for a performance evaluation of the (Gaussian process regression based) WLAN-only location estimator.) The large errors during the last minutes can be explained as follows. After a few seconds outside of the right building, GPS data are received and the filter integrates GPS observations into the solution. The error of the GPS stand-alone solution is still high but decreases as the solution converges to the true location. Simultaneously, while the mobile terminal departs further from the building, the radio map data becomes more ambiguous, which increases the WLAN-only error. The error of the hybrid solution increases because GPS data are still inaccurate, they are still converging to the true location, and

TABLE 1: Average RMS error and standard deviation of the particle filter in GPS + WLAN, GPS-only, and WLAN-only configuration and of the extended Kalman filter for Trajectory-1.

	PF, GPS + WLAN	PF, GPS-only	PF, WLAN-only	EKF, GPS + WLAN
RMSE (m)	6.21	37.33	7.80	12.97
Avg. SD (m)	5.81	48.27	6.02	2.35

FIGURE 4: Empirical error distribution function of the particle filter in GPS + WLAN, GPS-only, and WLAN-only configuration and of the extended Kalman filter for Trajectory-1.

WLAN data are becoming inexact. The particle filter weights the WLAN data more than the GPS data until minute 25. Then the GPS observations get more accurate and the particle filter begins weighting the GPS observations more than the WLAN observations. This is the desired effect; as one can see, during the last minute, the GPS-only solution yields higher accuracy than the WLAN stand-alone solution. Shifting the weights to GPS observation induces a decrease of the GPS + WLAN solution's error. Notice that the hybrid solution's error is always equal or lower than the error of the better performing solution, pointing to an adequate weighting of the two information sources. However, the GPS data, on which the hybrid solution is relying during the last minute, is still relatively poor, maybe because the filter has not yet converged after the huge deviations, or because the last part of Trajectory-1 is surrounded by trees.

The error of the extended Kalman filter is initially equal to that of the GPS-only configuration. It takes about three minutes until a WLAN position estimate is considered by the extended Kalman filter. In that moment, the error decreases to the magnitude of the error of the particle filter. While the particle filter solution mainly ignores the GPS information, the extended Kalman filter occasionally weights the GPS data and deviates from the quite accurate WLAN-only solution. A better performance can be seen at the end of the path, when GPS data quality increases due to better visibility of the satellites. The GPS measurements get more weight while the WLAN data is weighted less. During the last minute, the extended Kalman filter outperforms the particle filter.

The precision of the algorithm can be assessed from Figure 4, which illustrates the empirical cumulative distribution function of the errors. The hybrid solution presents the highest accuracy and precision, with a median accuracy of 5 m and an accuracy of 11 m at 90% probability.

The extended Kalman filter solution yielded an accuracy of 10 m at 50% and 20 m at 90% probability. That is about twice as low compared with the accuracy of the particle filter. The maximum error of the extended Kalman filter solution is about 30 m, whereas the maximum error of the particle filter is 15 m.

Table 1 compares the performance of the particle filter hybrid solution with the individual GPS and WLAN solutions and with the extended Kalman filter numerically, that is, in terms of the root-mean-square error (RMSE) and the standard deviation (SD). The hybrid solution clearly outperforms the two other particle filter configurations. The GPS-only solution is very inaccurate, because GPS observations were only available during the last section of the trajectory. For the same reason, the WLAN stand-alone solution is almost as good as the hybrid solution. Nevertheless, it has a higher probability for large errors. Due to the additional information in terms of GPS observations in the last section of the trajectory, the hybrid solution is more accurate than the WLAN-only solution. The RMSE of the particle filter WLAN + GNSS configuration is in comparison with the extended Kalman filter solution two times smaller. However, considering the standard deviation of these two filters, the outcome of extended Kalman filter is two times smaller than that of the particle filter.

The error distribution function and the standard deviation of the particle cloud show that the hybrid solution is not only more accurate but also more precise; its empirical error distribution function presents the lightest tails. The GPS + WLAN configuration also yielded the smallest standard deviation.

5.4. Outdoor-Like Trajectory: Trajectory-2. The overall performance of the filter for Trajectory-2 is depicted in Figure 5. The continuous availability of GPS observations is observable; the stop-and-go motion causes the position estimates to cluster, because, during the phases without motion, the filter continues estimating position of the static mobile terminal based on GPS observations. In between these clusters, the estimated position tends to jump, which can be explained again by the reception of a reliable RSSI observation after some time. After the estimates converge closer to the starting point, the estimated path follows roughly the ground truth. The estimates become more exact where the mobile terminal enters the right building. However, most times along the roofed passageways and around the small lawn area the localization performance is only fair. More exact estimates can be seen about the corner, where the two buildings almost touch, and later on the open space again.

The extended Kalman filter shows a similar positioning performance. It estimates well the entering of the right

FIGURE 5: Particle filter, extended Kalman filter estimates, and ground truth of Trajectory-2. Results of the particle filter are shown for the three configurations GPS + WLAN, GPS-only, and WLAN-only.

FIGURE 6: Error of the particle filter in GPS + WLAN, GPS-only, and WLAN-only configuration and of the extended Kalman filter for Trajectory-2.

building. On the semiopen passageway of the left building and around the small lawn area, the accuracy decreases as well; nevertheless, position estimates of the extended Kalman filter appear smoother than those of the particle filter. The accuracy of the extended Kalman filter estimates increase again towards the end of the path on the open space.

To reason about the causes of the described performances, we refer to Figure 6. It illustrates the magnitude error, again for the three configurations of the hybrid algorithm and the extended Kalman filter. This figure shows that the error of the particle filter GPS + WLAN solution is in average below 5 m; nevertheless, in various moments, it exceeds 10 m. The GPS-only solution yields errors similar to the hybrid solution and errors smaller than that of the WLAN stand-alone solution. The outcomes based on only WLAN are very poor, especially about 24 min to 26 min and for the last 4 min of the trajectory.

FIGURE 7: Empirical error distribution function of the particle filter in GPS + WLAN, GPS-only, and WLAN-only configuration and of the extended Kalman filter for Trajectory-2.

The low accuracy of the particle filter's GPS + WLAN solutions about 24-25 min is because of the poor quality of WLAN and GPS observations. During that time interval, the WLAN observations are very inexact for which reason they are basically unregarded, and therefore the accuracy of the hybrid solution equals that of the GPS-only solution. Between minutes 34 and 35, the poor accuracy of the hybrid solution is caused by an inappropriate weighting of the observations. The filter is too confident about the GPS observations or estimates the quality of the WLAN observations as too poor, respectively. The same aspect is visible at the beginning around 23 min and also about 27.5 min. In general, however, the accuracy of the hybrid solution, again, leans towards the more accurate sensor information; it never presents the largest errors among the three configurations. Figure 6 indicates that the weighting mechanism works reasonably well besides a few sporadic exceptions.

The accuracy of the extended Kalman filter is also most of the time about 5 m and shows only a few peaks that are larger than 10 m; see, for example, minutes 24, 28, and 34. It is worth noting that in some occasions the extended Kalman filter presents the lowest errors; refer to minutes 25, 35, 37, and 38. In general, Figure 6 confirms the similarity of the accuracy of the particle filter in GPS + WLAN configuration and the extended Kalman filter.

We examine again the cumulative distribution function of the error, as shown in Figure 7. The proposed algorithm yielded a median accuracy of 4.5 m and only 10% of the errors are larger than 10 m. As the hybrid solution makes use of both sensor data, it is not surprising that it outperforms the individual solutions. The median error for GPS-only is roughly 6.5 m and that of the WLAN-only configuration is 8 m. The difference is even bigger at 90% probability; GPS-only yielded 12 m and WLAN-only 23 m. Thus, in comparison with the two stand-alone solutions, the integration of the two systems improves also the precision.

The median accuracy of the extended Kalman filter is 5.5 m, only one meter larger than that of the particle filter hybrid solution. The error at 90% probability is, as for the particle filter in GPS + WLAN configuration, 10 m. The maximum error of the extended Kalman filter is about 18 m, that is almost 10 m less than the maximum error of the hybrid

TABLE 2: Average RMS error and standard deviation of the particle filter in GPS + WLAN, GPS-only, and WLAN-only configuration and of the extended Kalman filter for Trajectory-2.

	PF, GPS + WLAN	PF, GPS-only	PF, WLAN-only	EKF, GPS + WLAN
RMSE (m)	6.68	7.88	13.36	6.49
Avg. SD (m)	2.70	2.99	4.81	1.04

solution. However, the largest error of the particle filter stems clearly from the initial estimate, before the first observations were obtained; the maximum error of the extended Kalman filter occurred about 34 min.

Table 2 summarizes the performance analysis and confirms the previously made notions. It presents the RMS error and the standard deviation derived from the particle cloud and contrasts them for the three configurations, as well the RMSE and the standard deviation of the extended Kalman filter are shown. The extended Kalman filter yielded the smallest RMS error and standard deviation, although the GPS + WLAN configuration's median error is smaller than the median error of the extended Kalman filter. The hybrid solution of the particle filter is in average slightly more inaccurate and its standard deviation is more than twice than that of the extended Kalman filter. As expected, the particle filter in GPS + WLAN configuration outperforms the GPS-only and WLAN-only solutions. The GPS stand-alone solution performs also well considering the localization scenario. The performance based on only WLAN observations is mediocre, because large parts of the paths are outdoors, where fingerprints are ambiguous.

The stated error figures are specific for that system and scenario and shall be put into context. If the measurements of GPS and WLAN are severely degraded, the particle filter can only yield poor estimates. An idea about the quality of the measurements can be gained from Figure 8.

Figure 8 compares the estimates for Trajectory-2 of the GPS-only particle filter solution with that of the commercial GPS receiver ([22]) and with the least squares (LS) solution (that was used to estimate the clock offset of the receiver). Additionally, this figure depicts a second trajectory from the commercial GPS receiver from another experimental run. The black trajectory is derived from the same GPS observations that were fed into the particle filter (teal curve) and least square algorithm (cyan curve). The grey trajectory is an alternative trajectory estimated by the GPS receiver and was recorded on the same path, but during another experiment; the experimenter's motion during this experiment was continuous.

The commercial GPS receiver estimates (black path) the trajectory well at the beginning, until the path enters the right building. The opposite holds for the GPS-only solution of the particle filter; its estimates are pulled towards the open space. The indoor part is not recognized by the GPS receiver, whereas the outcomes of the GPS stand-alone solution are good in the indoor section. On the roofed passageway of the left building, only the GPS receiver provides more or less adequate estimates; the estimates appear of higher accuracy than that of the GPS-only solution. Later, on the passageway of the right building, the GPS receiver's accuracy decreases

FIGURE 8: Position estimates of the GPS-only solution, the GPS receiver, and the least squares solution. The black (GPS rcv.) and the teal trajectory (PF, GPS-only) were obtained from the same experimental run. The grey trajectory (GPS rcv.) was recorded during a different experiment along the same path, though, without the stops.

again; its estimates cross the right building until they come back on track which in the receiver is on the open area. The GPS-only solution shows better results on the passageway of the right building, but on the open space the GPS receiver appears to be more accurate again.

The least squares solution illustrates the quality of the GPS measurements and the alternative trajectory of the GPS receiver shows the harshness of the environment. The estimates of the least squares solution are similar compared to that of the GPS-only solution, but much more noisy due to the missing motion model. Nonetheless, position fixes during the indoor part are surprisingly good. The alternative GPS trajectory is worse than the black trajectory, especially on the semiopen passageway of the right building and after surrounding the small lawn area; no position fixes could be estimated during the last minutes of the trajectory.

A comparison of the accuracy of the trajectory of the GPS receiver and the hybrid particle filter is shown in Figure 9. That figure depicts the magnitude error of Trajectory-2 obtained from the hybrid particle filter next to the error from the GPS receiver. Additionally, we show the number

FIGURE 9: Magnitude error of hybrid GPS + WLAN solution and of GPS receiver.

of satellites. At the beginning of the trajectory, on the open space, the GPS receiver is more accurate than the hybrid solution. It actually follows the ground truth very well, even on the passageway of the right building, which is remarkable under these conditions (it might even be an exception as the alternative trajectory of the GPS receiver suggests). The indoor section and the path next to the left building, in general after minute 29, are nevertheless estimated poorly. Errors rise up to 20 m and 30 m. These large errors occur when the number of available satellites is varying a lot, indicating quick changes of the sky view which degrades GPS performance. Time intervals of good GPS performance correlate with sections where the number of observations is constant.

For most of the time of the trajectory, 6 to 7 GPS observations are available, a fair amount, given the scenario. Only in a few epochs, the number of satellites decreases to four and only one time it is below four. That means, that during time intervals of poor GPS performance, the number of observations was fair enough to get a good position fix, but that the quality of the observations was not sufficient for accurate position estimates.

The integration of WLAN data improves the localization accuracy. The GPS receiver's RMSE for that trajectory is 9.79 m, about three meters larger than the RMSE of the particle filter and of the extended Kalman filter.

6. Discussion

The performance of GNSS degrades in areas with obstructed satellite-to-receiver link, because, for example, scattering and multipath effects delay the signal and increase the propagation time. However, the same obstructions attenuate WLAN signals so that the spatial RSSI distribution is rather heterogeneous. This facilitates distinguishing nearby RSSIs and improves the resolution and the accuracy of a WLAN fingerprint localization system. The opposite effect occurs in open areas, where GNSS's position fixes are usually accurate, but position estimates relying on WLAN RSSI often show large errors. These large errors are caused by very similar RSSI patterns at distant locations.

As expected, the particle filter hybrid solution attains the highest accuracy and precision compared with the stand-alone solutions. The smaller error of the two stand-alone configurations presents an upper bound for the error of the hybrid solution.

In comparison with the extended Kalman filter, the proposed algorithms outperformed the extended Kalman filter in the indoor-like scenario. For Trajectory-2, the performance of the particle filter and the extended Kalman filter are comparable, the particle filter performed in average slightly worse than the extended Kalman filter. The extended Kalman filter estimates are more precise than the estimates of the particle filter. This is because of the little amounts of noise that are introduced in the process model of the particle filter to mitigate sample impoverishment (caused by resampling). However, the extended Kalman filter's performance is considerably poorer (large position jumps) than that of the particle filter when only WLAN data was available. The integration of WLAN data works better in the particle filter. From our point of view the disadvantage that the extended Kalman filter presents in scenarios without or with few GPS data does not outweigh the small advantage shown in the scenario where GPS and WLAN data were basically available all the time.

Sections of low accuracy of the hybrid solution are due to the loss of one of the systems or due to very poor quality of one of the observations in combination with sharp direction turns and/or changes of the sky view. As for most of the time one of the systems operates reasonably well, extreme deviations from ground truth, as it is common for fusion on a position level and can be seen for the extended Kalman filter for Trajectory-1, can be avoided, because the continuous adaption of the weights results in smooth location estimates, also during indoor/outdoor transitions.

The chosen motion model is very general and versatile and served well in our experiments. However, an adequate model must be chosen according to the final application.

One drawback of our method is the computational complexity of the RSSI interpolation. Predicting RSSIs requires the inversion of the covariance matrix, which has a dimension equal to the number of used particles. One way to reduce the computational costs is to reduce the number of particles. To compensate the loss of accuracy due to reduction of particles, a better importance density is advisable. For some scenarios, an extended Kalman filter solution can achieve similar or better performance at lower computational complexity. Our methodology used a maximum likelihood estimator to estimate a position from WLAN RSSI; this is based as well on the Gaussian process regression. Replacing the used WLAN position estimation with a simpler algorithm (e.g., k -Nearest Neighbour) would reduce the complexity of the extended Kalman filter further. The concrete impact of the above-mentioned changes on the positioning accuracy of the particle filter requires further research.

The ubiquity of the proposed method depends heavily on the dissemination of WLAN fingerprinting location systems, more precisely on the coverage with radio maps, and that they are accessible and compatible. For further thoughts on that we refer to [4]. Another point we are presuming is the accessibility to the pseudoranges of a GNSS receiver. However, most

off-the-shelf receivers do not provide pseudoranges yet. This is, however, changing right now as smartphones with access to raw measurement recently appeared on the market.

One objective of this study was the exploitation of the contrary information provided by pseudoranges on the one hand and RSSI on the other hand. We consider the automatic weighting of the different sensor information as functioning, but it is also apparent that it is improvable. Some relatively simple and straight forward ideas and changes in the presented system are likely to improve the localization accuracy.

The simplest change is the use of a multiconstellation, multifrequency receiver. This improves the coverage because the use of those receivers increases the number of receivable signals, and it improves the localization performance because a second frequency enables a more accurate correction of GNSS errors. Exploiting the carrier phase information would significantly decrease GNSS positioning error; nevertheless, such receivers are just starting to penetrate the market. Additional improvement is expected from multipath mitigation methods in the receiver, for example, at the correlator stage, if these are not yet implemented in the used receiver. Multipath mitigation based on the RSSI data is also worth investigating.

On the WLAN side, the most crucial issue is the choice of the variance in the RSSI likelihood function. It appears advisable to consider time-dependent variance estimates that express the uncertainty of the WLAN observations and not of the predictions of the Gaussian process model. Combining the currently used RSSI variance, containing spatial information, and a RSSI variance estimated from the observed samples, containing temporal information, improves very likely the weighting between the two information sources. Approaches to reducing the computational costs of the interpolation of WLAN RSSI are another open point.

Dependent on the use case, one could create fingerprints only on the paths. This indirectly includes information of the environment and improves the localization performance by removing potentially ambiguous fingerprints. This idea also translates to the interpolation with Gaussian processes, because the variance in areas without fingerprints (training data) will be higher.

7. Conclusion

This study presents a new algorithm combining GPS pseudoranges and WLAN RSSIs. The proposed algorithm achieves accurate and robust seamless localization as it exploits the complementary information contained in the sensory data. We developed a likelihood function that overcomes the drawback of the spatially discrete fingerprints by interpolating RSSIs through Gaussian process regression. The integration of the RSSI likelihood function and the pseudorange likelihood function into the particle filter yields a fine grained, automatic weighting of GPS and WLAN observations, which improves the accuracy and precision when compared with the stand-alone solutions using only GPS or WLAN data. This gain in accuracy and precision comes from an increase in computational complexity. The extended Kalman filter used

as benchmark, that integrates a WLAN position estimate with GPS pseudoranges, achieved in one scenario similar positioning performance compared to the proposed particle filter. However, without enough GPS data, the extended Kalman filter accuracy degrades much, which is why we consider the particle filter worth its complexity.

In realistic indoor/outdoor experiments, in an environment harsh for GNSS and WLAN fingerprinting localization, the proposed method achieved an average positioning accuracy about 7 m. The proposed integration approach of GPS pseudoranges and WLAN RSSI is of low-cost and can improve localization performance of mobile platforms, whenever pseudoranges and RSSIs are available in the navigation processor.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was partially supported by the Consejo Nacional de Ciencia y Tecnología and the División de Investigación y Posgrado de la Facultad de Ingeniería de la Universidad Autónoma de Querétaro. The authors thank Karla Alejandra Rojas Camargo for the assistance in programming the data logger and data acquisition. Philipp Richter would also like to thank Eduardo Castaño-Tostado for his comments that improved this work.

References

- [1] E. D. Kaplan and C. J. Hegarty, Eds., *Understanding GPS: Principles and Applications*, Artech House Publishers, 2nd edition, 2006.
- [2] N. Samama, "Global Positioning: Technologies and Performance," *Global Positioning: Technologies and Performance*, pp. 1–419, 2007.
- [3] N. El-Sheimy and C. Goodall, "Everywhere navigation integrated solutions on consumer mobile devices," *Inside GNSS*, vol. 6, no. 5, pp. 74–82, September 2011.
- [4] P. Richter, M. Toledano-Ayala, G. M. Soto-Zarazúa, and E. A. Rivas-Araiza, "A survey of hybridisation methods of GNSS and wireless LAN based positioning system," *Journal of Ambient Intelligence and Smart Environments*, vol. 6, no. 6, pp. 723–738, 2014.
- [5] S. Zirari, P. Canalda, and F. Spies, "WiFi GPS based combined positioning algorithm," in *Proceedings of 2010 IEEE International Conference on Wireless Communications, Networking and Information Security (WCNIS)*, pp. 684–688, Beijing, China, June 2010.
- [6] D. Fernandez, F. Barcelo-Arroyo, I. Martin-Escalona, M. Ciurana, M. Jofre, and E. Gutierrez, "Fusion of WLAN and GNSS observables for positioning in urban areas: The position ambiguity," in *Proceedings of 16th IEEE Symposium on Computers and Communications, ISCC'11*, pp. 748–751, grc, July 2011.
- [7] Y. Ma, X. Chen, and Y. Xu, "Wireless local area network assisted GPS in seamless positioning," in *Proceedings of 2012*

- International Conference on Computer Science and Electronics Engineering, ICCSEE 2012*, pp. 612–615, chn, March 2012.
- [8] K. Nur, S. Feng, C. Ling, and W. Ochieng, “Integration of GPS with a WiFi high accuracy ranging functionality,” *Geo-Spatial Information Science*, vol. 16, no. 3, pp. 155–168, 2013.
 - [9] B. Li and K. O’Keefe, “WLAN TOA ranging with GNSS hybrid system for indoor navigation,” in *Proceedings of In Proceedings of the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation*, pp. 416–425, 2013.
 - [10] P. Richter, J. Seitz, L. Patiño-Studencka, J. G. Boronat, and J. Thielecke, “Wi-Fi azimuth and position tracking using directional received signal strength measurements,” in *Proceedings of 2012 Workshop on Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pp. 72–77, Bonn, Germany, September 2012.
 - [11] G. Hejc, J. Seitz, and T. Vaupel, “Bayesian sensor fusion of Wi-Fi signal strengths and GNSS code and carrier phases for positioning in urban environments,” in *Proceedings of 2014 IEEE/ION Position, Location and Navigation Symposium, PLANS 2014*, pp. 1026–1032, usa, May 2014.
 - [12] P. Richter and M. Toledano-Ayala, “Revisiting gaussian process regression modeling for localization in wireless sensor networks,” *Sensors (Switzerland)*, vol. 15, no. 9, pp. 22587–22615, 2015.
 - [13] C. E. Rasmussen and C. K. I. Williams, *Gaussian Process for Machine Learning*, The MIT Press, Boston, Mass, USA, 2006.
 - [14] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*, Cambridge University Press, New York, NY, USA, 2003.
 - [15] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, 2003.
 - [16] J. Vermaak and A. Blake, “Nonlinear filtering for speaker tracking in noisy and reverberant environments,” in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2001 (ICASSP’01)*, pp. 3021–3024, USA, May 2001.
 - [17] T. Takasu, *RTKLIB Manual*, 2013.
 - [18] Y. Bar-Shalom, X. Rong Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, Inc., 2001.
 - [19] M. Weyn, *Opportunistic seamless localization [Ph.D. thesis]*, 2011.
 - [20] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*, vol. 4, John Wiley & Sons, Inc, 2012.
 - [21] P. Richter, A. Pena-Torres, and M. Toledano-Ayala, “A rigorous evaluation of Gaussian process models for WLAN fingerprinting,” in *Proceedings of International Conference on Indoor Positioning and Indoor Navigation, IPIN 2015*, can, October 2015.
 - [22] u. blox, *u-blox 6 Receiver Description*, 2013.

Hindawi

Submit your manuscripts at
<https://www.hindawi.com>