UNIVERSITY OF BERN

BACHELOR THESIS

Kalman Filter supported WiFi and PDR based indoor positioning system

Author:
Matthias BACHTLER

Supervisor:
José CARRERA

Professor:
Prof. Thorsten Braun

Communication and Distributed Systems
Institute of Computer Science

February 7, 2018

UNIVERSITY OF BERN

Institute of Computer Science

Bachelor of Science

Kalman Filter supported WiFi and PDR based indoor positioning system

by Matthias BACHTLER

Abstract

This work implements an indoor localization system by fusing radio and pedestrian dead reckoning information in a Kalman filter approach.
Our localization approach has been tested in a complex office-like indoor environment. Experiment results show that this approach can achieve an average error of 3.2m and 90% accuracy of 4.1m. Compared to a PDR-based localization approach, our localization method outperforms it by around 60%.
Furthermore, the presented Kalman filter-based approach was compared to a particle filter-based localization system. While the particle filter system achieved a 3x higher localization accuracy, the required computational effort was 9x and the battery consumption 2x higher than with the Kalman filter system.
These findings suggest that the use of a Kalman filter may be of advantage, if system resources are limited and the localization accuracy requirements are not that strict.

Contents

List of Figures

List of Tables

List of Abbreviations

AP	Access Point
CDF	Cumulative Distributed Function
HTC1	HTC One, see section 4.1
ILS	Indoor Localization System
IMU	Inertial Measurement Unit
PDR	Pedestrian Dead Reckoning
RFID	Radio Frequency Identification
RSS	Received Signal Strength
SD	Standard Deviation
UWB	Ultra-wideband
Z1C	Sony Xperia Z1 Compact, see section 4.1

1 Introduction

Humans always wanted to know their own position. For instance, in ancient times, some landmark information, sun and other star knowledge were used for navigation, without or with the aid of tools such as compass or sextant and other similar devices. Nowadays, determining the outdoor position with a high accuracy is easy using a satellite-based navigation system, such as the Global Positioning System (GPS), or the cell phone network [21]. Accurate indoor localization on the other hand is less trivial. Using GPS usually is not possible due to signal attenuation of buildings, making the localization less accurate or impossible. Therefore, other technologies are required, such as trilateration using WiFi received signal strength information. A non-comprehensive overview of such techniques and technologies is given in section 2.1.

1.1 Motivation

Multiple indoor applications depend on an accurate determination of the position of material or persons, for example [38]:

1. Guidance
   - To guide users in large public buildings, such as shopping malls, airports, convention halls, libraries [1, 27]
   - To guide drivers to free parking spots [20]
   - Guidance or localization of robots in industrial or private settings, such as robotic vacuum cleaners
2. Monitoring of material or persons
   - To monitor material in an industrial setting, valuables during transport or to keep track of assets in complex storage situations [46].
   - To assist evacuation and rescue operations for firefighters [41], to monitor the locations of security guards [24], to ensure safety of miners in longwall coal mining [14], or to monitoring the location of employees in an office environment (Active Badge system) [49].
   - To quickly locate healthcare staff in an emergency, ensure safety of Alzheimer's or dementia patients, monitoring nursing time (time a nurse spends in patient room), tracking patient flow to find bottlenecks and monitor solutions, improve overall efficiency [7, 22, 23].
3. Location-aware applications
   - Indoor location-aware applications such as providing information, advertisement or discounts in shops or large shopping centres [36];
   - As a necessary tool to support augmented- or virtual-reality scenarios [26]

Of course, this classification is not absolute and combined services are imaginable; such as a multi-purpose museum app: It may guide visitors and display location-aware information about the exhibition. Furthermore, the museum may gather data about visitor movements, therefore recognizing and solving visitor flow bottlenecks.

In this work, a combined approach based on WiFi and PDR was chosen. These technologies work complementarily. PDR-based localization provides almost continuously information about relative movements, but can quickly lead to error accumulation. WiFi-based localization on the other hand helps countering the error accumulation by providing absolute localization information, but lacks the real-time ability.
Both types of information need to be fused. Kalman filter was chosen since it is computationally less demanding compared to other possibilities, such as the particle filter.

1.2 Contributions

In this work we propose an indoor localization system, which fuses radio and inertial measurements information in a Kalman filter approach.
The main contributions are as follows:

- We implemented a Kalman filter approach to achieve high indoor localization accuracy in smartphones.
- We conduct extensive experiments in an office-like scenario to demonstrate the performance of the presented system compared to PDR- and WiFi-based approaches.
  Experiments demonstrated a 90% accuracy of 4.1m. This outperforms the WiFi-based approach by 25% and the PDR-based approach by roughly 60%.
- We conduct experiments to compare the performance of the presented localization approach to a particle filter localization method. Performance is measured by comparing computational effort, battery consumption and localization accuracy.
  The comparison showed that the presented implementation only results in about 11% CPU usage and 50% battery consumption as compared to the particle filter implementation.

1.3 Overview

The remainder of this document is structured as follows: The theoretical background is presented in chapter 2, whereas chapter 3 describes the architecture of the implemented localization system. Chapter 4 provides implementation details. The performance of the localization system is presented in chapter 5. Chapter 6 concludes this work.

2 Theoretical Background and Related Work

This chapter introduces the theoretical background of the thesis: First, section 2.1 gives an overview of the work related to this thesis. Section2.2 introduces the localization approach based on WiFi Received Signal Strength readings. Section 2.4 elaborates the data fusion algorithm used in this work.

2.1 Related Work

Several technologies have been used to establish indoor localization systems (ILS); some of them specifically developed for this purpose, some of them are adapted from other applications. They can broadly be classified in 3 groups (introduction is based on the surveys of Brena and Gu [8, 19]:

1. Radio-based
   In radio-based localization systems, the mobile node communicates with several anchor nodes to determine the current position using two different groups of techniques:

   - Trilateration
     Trilateration is the process of calculating a position based on the distance of the mobile node to multiple other known locations. Figure 2.1 shows a trilateration situation with a position defined by the ranges to 3 anchor points. First, the distances of the mobile node to the anchor points in range are calculated using the measured RSS values according to a mathematical model (such as the lognormal channel model). In a second step, the current position is estimated according to a trilateration algorithm using the calculated ranges and the known positions of the anchor nodes.
     Other signal-related factors such as time-of-arrival or angle-of-arrival may be used instead of distances. Due to the short distances and the speed of light, a highly precise, time synchronized system is needed, if time-of-arrival is used.
     Localization techniquest based on trilateration are range-based. In contrast, fingerprinting-based and most of the following techniques are range-free, therefore do not depend on the calculation of distances to some known positions.
   - Fingerprinting
     During an extensive training phase, RSS values from multiple anchor points are gathered at different locations in the area to monitor. These RSS value sets are saved with the respective recording position in a database. The mobile node then measures the current fingerprint and compares it with the ones stored in the database to estimate its current position.

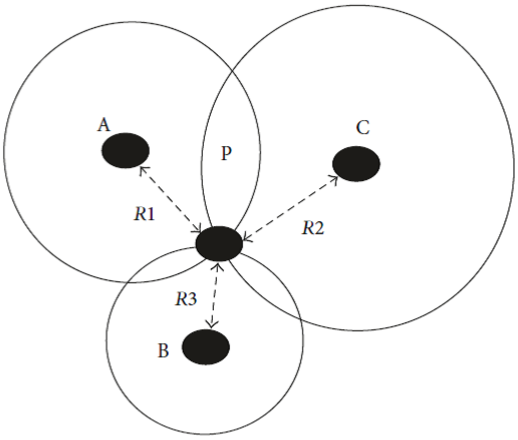


FIGURE 2.1: Schematic representation of a trilateration-situation. Ranges 1-3 to anchor point A-C are used to calculate the position of the mobile node [8].

> Usually, WiFi or Bluetooth signals are used, but the same principle may also be applied to other signals, such as magnetism.

Several pre-existing technologies such as WiFi, Bluetooth or ZigBee can be used. While WiFi is ubiquitously present to grant internet access and therefore need not to be installed additionally, its power consumption is higher that Bluetooth or ZigBee. Bluetooth on the other hand has very low power consumption, but the fixed beacons need to be installed specifically for this purpose. The iBeacon system by Apple is probably the most prominent Bluetooth-based system, but other have been developed as well [13]. It is used in Apple stores to show proximity-guided product information.

Both WiFi and Bluetooth have the advantage that smartphones directly can act as mobile nodes, since they already contain the necessary hardware, but the achievable accuracy is lower than with specialized systems.

Another wireless communication standard, ZigBee, a low-cost, low-data-rate and low-power, needs special hardware and is mainly used in smart homes, traffic light controlling and comparable use cases. ZigBee data packages already include the RSS values and therefore allow for a simpler implementation in the mobile node [52].

A more specialized radio-based technology is based on radio frequency identification (RFID): Usually passive tags contain information that can be read by a stationary or mobile reader. These tags are very cheap and do not need a power source (although active tags exist) and are therefore suitable to be widely distributed, either to tag anchor nodes or the mobile node. Usually, RFID-based systems do not monitor the position continuously but indicate the proximity to some predefined check points like doors [51]. However, more accurate systems with “real-time” positioning have been developed based on signal intensity and fingerprinting, using the k-nearest neighbour algorithm to find the position [40].

A very interesting application using RFID positioning is the SeSaMoNet: Buried RFID-chips which can be read by a reader mounted on the tip of a walking stick guide the visually impaired via a smartphone and headset. This system connects the railway station of Laveno (northern Italy) with the shore of Lake

Maggiore [6, 11]. This guiding system was created in an outdoor scenario, but the same technique could be applied indoor as well.
The last radio-based technology applied in indoor localization is ultra-wideband (UWB). It offers advantages in multipath immunity and low-power requirements, but requires dedicated infrastructure and devices. A published system consisting of fixed transmitters and mobile receivers showed an accuracy of 1m [4].

2. Environmental-based
Several natural or artificial environmental factors can be used to estimate the current position: Magnetic field, background noise or light. Using background information without embedded signals, detailed fingerprint databases need to be established. Furthermore, the corresponding parameter should not vary over time or the variation needs to be covered by the data set, thus enlarging it. Such fingerprinting approaches may not be very precise, but allow the localization at room level. Furthermore, extensive calibration at different time points (for ambient light, summer versus winter, day versus night) is needed. The use of multiple environmental factors may increase the accuracy [15].
Localization based on light and sound can be supported by introducing position-specific information via light bulbs or speakers. This information is not visible or audible for humans but can be detected by smart phones or other specialized sensor devices. In this case, no fingerprint databases are necessary. The localization is determined like with radio-based techniques.
A highly accurate system based on light bulbs emitting location information showed an accuracy of 6cm [54]. Of course, this increased accuracy is paid by the necessity of specialized hardware, therefore increasing the cost of the localization system.

3. Other localization technologies and techniques do not fit into the aforementioned categories, such as:

   - Dead Reckoning
     The current position is based on an initially known position, speed and direction of movement. Measurements based on accelerometers, gyroscopes and magnetometers enable the mobile node to calculate the direction of movement and step detection to estimate movement speed. Since the measurements are not very accurate, errors accumulate and can get substantial, if the positioning runs for some time. Therefore, these systems are usually coupled with other technologies to update the current position and therefore reduce the error accumulation.
     As an example, Beauregard and Haas used a specialized, head-mounted motion sensors to achieve highly accurate results. Even the authors admitted that simpler sensor options such as smartphone sensors would only result in coarse location information [5]. Dead reckoning, or inertial navigation, is widely used in marine, air and space traffic.
   - Vision analysis
     Analysis of images gathered by either wall- or user-mounted cameras enables the localization system to estimate the location of the mobile node [55].
     A specialized application of a vision analysis system is the Microsoft Kinect system, which tracks player movements to control games [33].

- Ultrasound-based systems
  The mobile node carries an emitter of ultrasound signals, which are received by multiple anchor nodes. The system then calculates the position of the mobile node similarly to some radio-based approaches: Based on differences in arrival times at different anchor nodes and multilateration. Because the localization calculations are performed centrally, the locations of all mobile nodes are known by the central system. Depending on the privacy requirements of the application, this may be an issue. An implementation of this principle, fittingly called bat system, achieves the positioning with an accuracy of a few centimetres. Although highly accurate, the system uses specialised hardware and therefore cannot be used ubiquitously [50].

4. Combinations
   Usually, not a single technology is used in an ILS, but a combination of techniques and technologies.
   Closely related to this work are ILS based on WiFi and PDR, such as the work of J.Carrera [9]. They established a WiFi- and PDR-based localization system, using a particle filter as a fusion algorithm. In contrast, the work by Tarrío and colleagues, which serves as a basis for this thesis, used a Kalman filter for data fusion [48].
   Other groups implemented similar systems, based on WiFi or Bluetooth, usually supported by PDR. Data is either fused using particle filters or some form of Kalman filters. Combinations of both exist: A Kalman filter is responsible to smoothen noisy sensor input, while a particle filter performs the actual localization. The systems differ in details, such as ranging method or trilateration algorithm used [30, 25, 34, 42].
   Others included the use of landmarks, such as doors, stairs and elevators, to reset PDR localization and therefore limit the error accumulation [10].
   All these indoor localization systems utilize ubiquitously present technologies, but do provide only moderately accurate localization information (1-3m error). Using other technologies, reaching higher accuracy is well possible, but usually requires special hardware. Some examples are mentioned in the beginning of this section.

2.2 WiFi-Based Localization

The localization approach based on WiFi received signal strength (RSS) readings consist of two methods: Ranging, introduced in sub-section 2.2.1 and trilateration, in sub-section 2.2.2.

2.2.1 Ranging

Radio signals get attenuated while propagating through space. To establish a relationship between RSS and the distance between sender and receiver, several models have been described, most notably the lognormal channel model [44]. This model does not consider modifying effects such as multipath propagation or shadowing effects and is therefore not very precise in indoor situations.
For ranging, the calculation of the distance from the mobile node to the anchor

nodes, this implementation uses the following nonlinear regression model to calculate the distance d_i to the ith anchor node, based on the received WiFi signal strength (RSS) [35]:

$$d_i = \alpha_i * e^{\beta_i * RSS_i}, \tag{2.1}$$

where α_i and β_i are experimentally determined anchor-node specific parameters, and RSS_i, the WiFi signal strength received from the anchor node i.
The collection of data and the necessary calculations to determine α's and β's is presented in detail in chapter 5.1.1.

2.2.2 Trilateration: Hyperbolic Positioning Algorithm

To calculate the current position based on the distances to the WiFi access points (trilateration), the weighted hyperbolic positioning algorithm is used according to the following formula [47]:

$$\bar{x}_{RSS} = (H^T * S^{-}1 * H)^{-}1 * H^T S^{-}1\tilde{b}, \tag{2.2}$$

where the matrix H contains the positions of the anchor nodes (AN), the vector $\tilde{b}$ and the weighting matrix S. Weighting is finally based on the distance to each anchor node, therefore anchor nodes which are further away have a lower influence on the resulting position.
The matrices are defined as follows:

$$H = \begin{bmatrix} 2x_2 & 2y_2 \\ \vdots & \vdots \\ 2x_N & 2y_N \end{bmatrix}, \tag{2.3}$$

$$\tilde{b} = \begin{bmatrix} x_2^2 + y_2^2 - \tilde{d_2}^2 + \tilde{d_1}^2 \\ \vdots \\ x_N^2 + y_N^2 - \tilde{d_N}^2 + \tilde{d_1}^2 \end{bmatrix}, \tag{2.4}$$

$$S = \begin{bmatrix} \tilde{d_1}^4 + \tilde{d_2}^4 & \tilde{d_1}^4 & \dots & \tilde{d_1}^4 \\ \tilde{d_1}^4 & \tilde{d_1}^4 + \tilde{d_3}^4 & \dots & \tilde{d_1}^4 \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{d_1}^4 & \tilde{d_1}^4 & \dots & \tilde{d_1}^4 + \tilde{d_N}^4 \end{bmatrix}, \tag{2.5}$$

where $\tilde{d_i}$ is the distance to the anchor node i and x_i and y_i are the anchor node's x- and y- coordinates.

Without loss of generality, all anchor nodes are translated in order to set the first anchor node to $(0/0)$ and therefore to simplify the formula.

There are more precise positioning algorithms such as the (weighted) circular positioning algorithm or combined algorithms [47, 35]. The current algorithm was chosen due to its low computational requirements, which is one of the Kalman filter's advantages.

2.3 PDR-Based Localization

Inertial measurement units (IMUs) such as accelerometer and magnetometer provide information about movement of the mobile node. Therefore, only relative position changes can be derived, but no absolute position information.

2.4 Data Fusion

Several data fusion algorithms exist. In this work, a Kalman filter as described in subsection 2.4.1 is used, due to its low computational demand. This algorithm will be compared to a particle filter-based approach. Therefore, a short introduction to particle filters is given in subsection 2.4.2.

2.4.1 Kalman Filter

Kalman filter is a very popular, widely used recursive algorithm for data fusion or filtering, published by Kálmán in 1960 [29].
It is an estimation algorithm used to estimate the state of a linear system, such as the position and velocity of a vehicle. Figure 2.2 shows the 3 phases of a Kalman filter: Prediction, measurement and update. During the prediction phase, the current state is predicted according to the system's state model, and based on the previous system state; possibly modified by control inputs. To verify the prediction, a measurement of the system state is performed. The new system state is then updated by a linear combination of the measurement and the prediction, weighted by further information relating to the process and measurement error.

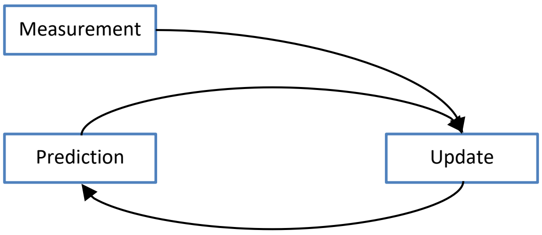


FIGURE 2.2: The Kalman filter algorithm calculates the future state based on the current state and the underlying model. This prediction is then updated using some sort of measurement input. The influence of the measurement correction is based on the process and measurement noise covariance matrices, respectively.

It is a Bayesian estimation method and provides the optimal estimate for linear system models (if the system and measurement noise is additive and independently distributed with a zero-mean).
One of the most prominent usages of a Kalman filter is probably the application in the Apollo Space program to estimate the position and velocity of the space crafts, but it was widely used in naval, aerial and space flight navigation and calculations [39].
Today it plays a central part in navigation systems by providing smoothed satellite navigation solutions, calibration of inertial navigation systems and integration of various information sources [18]. More specialized applications, such as a modelling an oil-fired power plant [43] or the improvement of dialysis quality have been

published [2, 32].
Although the initial Kalman filter was developed to solve linear problems, variants to handle non-linear problems were derived, such as the extended or the unscented Kalman filter [28, 53].
The remainder of this section presents each phase of the Kalman Filter in detail: First the general case formulas are presented, followed by the formulas specifically used in this work.
The overview is adapted from lecture notes by R. Faragher and a book by P. Groves [12, 18].

State Model

The state of a linear system at time t can be described by the state vector $\hat{x}_t$, a set of parameters describing the system.
Since it is a recursive algorithm, it depends on the state of the system at time $t-1$:

$$\hat{x}_{t|t-1} = F_t x_{t-1|t-1} + B_t u_t + w_t, \tag{2.6}$$

where F_t is the state transition matrix describing the transition of the state from time $t{-}1$ to time t; the control input matrix B_t which describes how the control input parameters in the vector u_t modulate the system state; and finally a process noise vector w_t with terms for each parameter of the state vector with error covariance matrix Q_t (used later in the prediction phase, see 2.4.1). This matrix represents the uncertainties in the state estimates and the degree of correlations between the errors.

In this case, no transition matrix is necessary. The system is controlled by the input matrix and parameter:

$$\hat{x}_t = \begin{bmatrix} x_t \\ y_t \end{bmatrix} = \hat{x}_t + \Delta T_{WiFi} * \begin{bmatrix} v_t * \cos\theta_t \\ v_t * sin\theta_t \end{bmatrix}, \tag{2.7}$$

where ΔT_{WiFi} is the time between WiFi signal strength updates, the velocity $v_t = (steplength)/(\Delta T_{step})$ and the time between steps ΔT_{step}. θ_t is the current heading of the mobile node, collected by the compass module (4.2).

Covariance matrix Q_t, which describes the process noise $w_t = \Delta T_{WiFi} * \Delta v_t + 1/(2 * \Delta T^2 * \Delta a_t)$, is constituted as follows:

$$Q_t = I * (\frac{1}{2\sigma_a \Delta T_{WiFi}^2})^2 + I * (\sigma_v \Delta T_{WiFi})^2, \tag{2.8}$$

where I is the identity matrix and σ_a and σ_v are the standard deviations of acceleration and speed, respectively.

Prediction Phase

The next system state $\hat{x}_{t|t-1}$ and the corresponding covariance matrix $P_{t|t-1}$ are predicted based on the previous system state at time t-1, possibly modified by control inputs:

$$\hat{x}_{t|t-1} = F_t x_{t-1|t-1} + B_t u_t, \tag{2.9}$$

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t, \tag{2.10}$$

where Q_t is the process noise covariance matrix (associated with noisy control inputs).
As already shown in section 2.4.1, the next state is defined as follows:

$$\hat{x}_{t|t-1} = \hat{x}_{t-1|t-1} + \Delta T_{WiFi} * \begin{bmatrix} v_t * \cos\theta_t \\ v_t * sin\theta_t \end{bmatrix}, \tag{2.11}$$

with the covariance matrix

$$P_{t|t-1} = P_{t-1|t-1} + Q_t, \tag{2.12}$$

where v_t is the velocity and θ_t the heading at time t. This part of the system is implemented using pedestrian dead reckoning. Therefore, the velocity and the heading are provided by the PDR module.

Measurement Phase

On the other hand, the state of the system is determined by a measurement, described by the measurement vector z_t:

$$z_t = H_t x_t + v_t \tag{2.13}$$

and the covariance matrix

$$R_t, \tag{2.14}$$

where H_t is a transformation matrix that maps the state vector parameters into the measurement domain and the measurement noise vector v_t with terms for each observation in the measurement vector. The covariance matrix R_t describes the measurement errors and will be used in the update phase (see section 2.4.1).

In this case, the current system state is determined by ranging (2.2.1) and trilateration 2.2.2, yielding the WiFi-based position $\bar{x}_{WiFi}$:

$$\bar{z}_t = \bar{x}_{WiFi} = (H^T * S^{-1} * H)^{-1} * H^T S^{-1} \tilde{b}, \tag{2.15}$$

where the matrices are defined as shown in section 2.2.2. Measurement error is described by the measurement noise covariance matrix R_t:

$$R_t = I * \sigma_p^2, \tag{2.16}$$

where I is the identity matrix and σ_p is the position's standard deviation.

Update Phase

Measured and predicted system state are then combined to the updated state $\hat{x}_{t|t}$ by correcting the predicted state by $K_t * y_t$:

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t y_t, \tag{2.17}$$

$$y_t = z_t - H_t \hat{x}_{t|t-1}. \tag{2.18}$$

The covariance matrix P_t is updated in a similar manner:

$$P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1}, \tag{2.19}$$

$$K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1}, \tag{2.20}$$

where y_t is the Innovation, K_t the Kalman gain which optimally weights the update according to the uncertainty of the current state estimates, and R_t the measurement noise covariance matrix.
Innovation is a measure for how well the measurement and the prediction agree with each other. If the innovation is large, they do not agree well and the prediction will be corrected accordingly, weighted by the Kalman gain.

The predicted position is updated with the measured position $\bar{x}_{WiFi}$:

$$\hat{x}_{t|t} = \hat{x}_{t-1|t-1} + K_t * \tilde{y}_t, \tag{2.21}$$

where the innovation $\tilde{y}_t$ is calculated as follows:

$$\tilde{y}_t = \bar{x}_{WiFi} - \hat{x}_{t|t-1}, \tag{2.22}$$

with Kalman gain defined as

$$K_t = P_{t|t-1} * S_t^{-1}, \tag{2.23}$$

with the innovation covariance matrix S_t:

$$S_t = P_{t|t-1} + R_t. \tag{2.24}$$

Finally, the covariance matrix $P_{t|t}$ is updated as well:

$$P_{t|t} = (I - K_t) P_{t|t-1}. \tag{2.25}$$

2.4.2 Particle Filters

Particle filters or "sequential Monte Carlo" methods are used to estimate the state of a dynamic system, such as the position of a mobile node. Like Kalman filters, they are recursive Bayesian filter, based on predict-update cycles. Instead of describing the probability density function in a functional form, it is approximated by a set of random samples of the density function, the particles. By increasing the number of those particles, the approximation can be made as accurate as desired.
Advantages of particle filters are their scalability towards high-dimensional problems and by changing the number of particles, accuracy and computational costs can be balanced. Furthermore, they can easily include additional information, such as floor plan constraints. However, they are usually computationally more expensive [45].

3 Localization System Architecture

This chapter describes the architecture of the localization system. It consists of 3 phases:

1. Calibration
   The calibration phase is presented in more detail in section 3.1.

2. Localization
   The architecture of the localization module is explained in section 3.2.

3. Analysis
   Analysis of the data, which is collected during a localization experiment, is explained in section 3.3.

Section 3.4 concludes this chapter with a short architectural comparison between this work and the particle filter approach of Carrera et al. [9].

3.1 Calibration Phase

During an offline calibration phase, the calibration parameters used in the ranging process are gathered, as presented in figure 3.1:

1. Calibration Data Collection
   At several calibration points per room, the RSS values of each access point were recorded using the localization application. The position of each calibration point was determined using a Disto D5 laser distance meter (Leica Geosystems). The positions of the calibration points are shown in figure 5.2 as blue crosses.

2. Calculation of Ranging Parameters
   Calibration data was analysed on a computer using Microsoft Excel 2013 by plotting the distance against the mean of 5 consecutive RSS values. The access point- and cell phone-specific ranging parameters α_i and β_i were calculated by exponential curve fitting by Microsoft Excel. Figure 5.4 shows one calibration data set, table 5.1 the ranging parameters found.

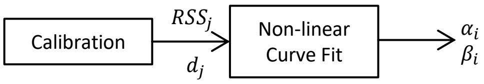


FIGURE 3.1: At selected calibration points, the RSS values and the distances to all access points are recorded. These data is used to calculate ranging parameters.

3.2 Localization Module Architecture

The architecture of the presented localization system is shown in figure 3.2. The system contains 4 main elements:

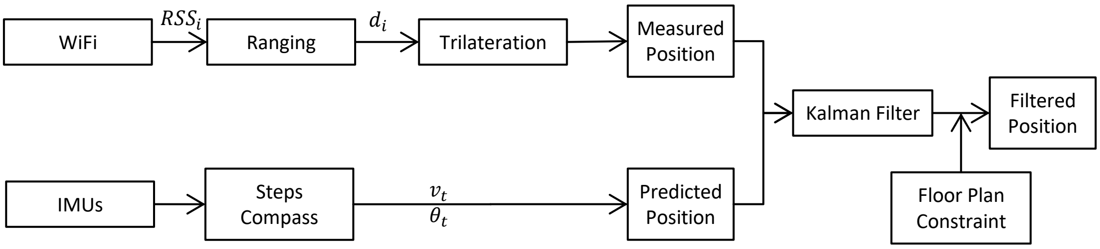


FIGURE 3.2: The upper arm represents the position measurement using WiFi received signal strength, the lower arm the PDR-based localization.

1. WiFi received signal strength measurements
   Based on calibration parameters α_i and β_i, WiFi received signal strength measurements are used to calculate the distance from the mobile node to each anchor node (ranging, see section 2.2.1). Based on these distances, the current position of the mobile node is calculated as described in section 2.2.2.

2. Pedestrian Dead Reckoning
   IMUs provide direction θ_t and velocity v_t of movement. These parameters predict the next position.

3. Kalman Filter
   The Kalman filter fuses the output of WiFi- and PDR-based localizations, as described in section 2.4.1.

4. Floor Plan Constraint
   The output of the Kalman filter is tested for a violation of the floor map constrains. If the position violates the constraint, the position is corrected. Figure 3.3 shows the principle of the constraint testing and possible correction of position. Otherwise, map information is not used in the localization system, since Kalman filter algorithm itself is not designed to use additional, map-based information.

3.3 Data Analysis

This section describes the data flow in the presented localization system.

1. Localization Data Collection
   During the localization experiments, the user saves the measured position of the mobile node together with the real position using the localization software (section 4.2) on the mobile node.
   The measured position is shown on the map during the experiment (see figure 4.1).

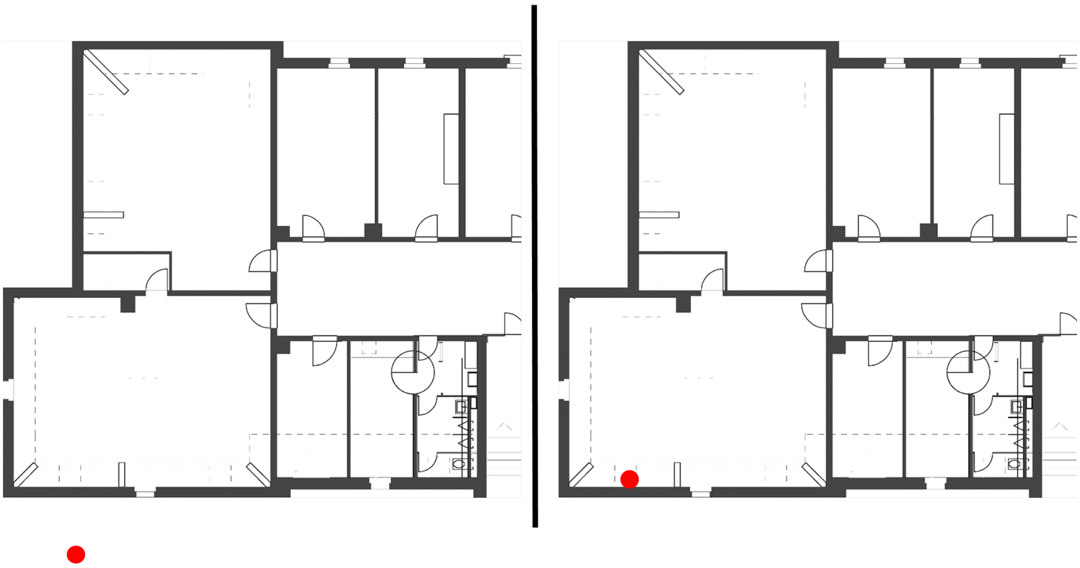


FIGURE 3.3: Floor plan-imposed constraint: If a position is found to be outside of the map (left), it gets moved to the closest possible position within the experimental area (right).

2. Localization Data Analysis
   After transferring the saved experiment files to the analysis computer, the distances of the measured positions to the corresponding real positions are calculated according to equation 5.1 with Microsoft Excel 2010. This distance is the measurement error.

3. Visualization
   Results are visualized either using Microsoft Excel or GNU Octave

3.4 Comparison with Particle Filter Application

The results of this work will be compared with a particle filter based approach by Carrera et al. [9]. Therefore, its structure is shortly discussed here, as shown in figure 3.4.
This localization system is also based on PDR, WiFi RSS measurements and ranging. These measurements are further supported by floor map constraints. While the system presented in this work uses a floor map only to correct positions outside of the experiment area, the system of Carrera et al. excludes particles in restricted areas. This does not only make positions outside of the are impossible, but also eliminates positions in walls. The most important difference is surely algorithm used to fuse WiFi and PDR information: Carrera implemented a particle filter, while in this work a Kalman filter was used.

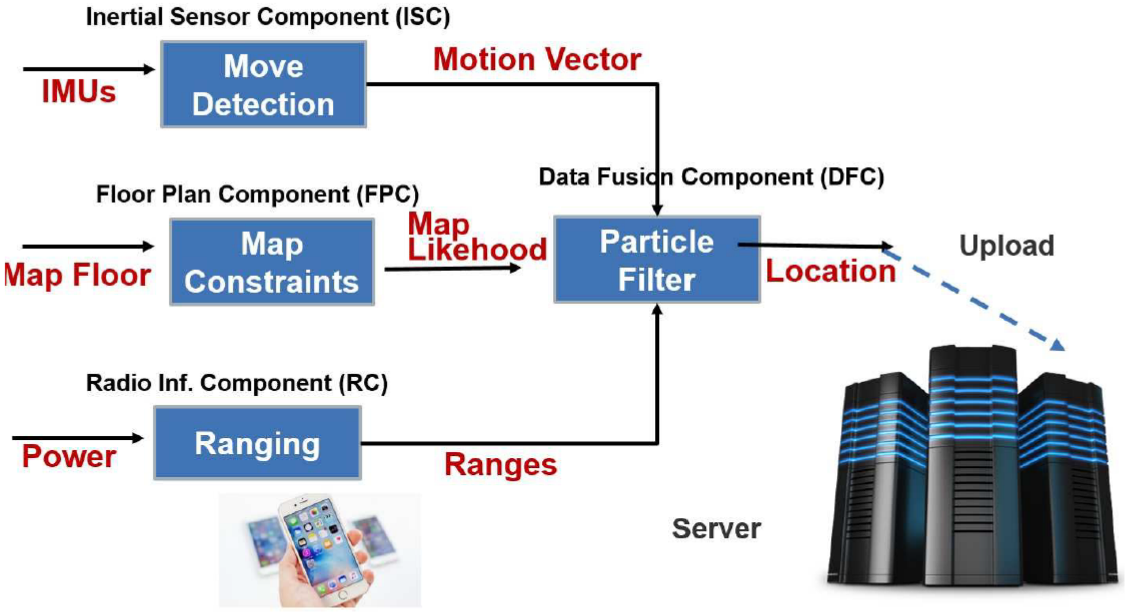


FIGURE 3.4: Particle Filter Implementation Structure [9].

4 Localization System Implementation

This chapter presents the implementation of the ILS: First, the mobile and anchor nodes are described in section 4.1 and the implemented software is presented in section 4.2. Finally, the evaluation scenario is described in section 4.3.
The implemented localization system consists of 3 main parts:

- Mobile nodes
  Cell phones are used as mobile nodes.
- Anchor nodes
  WiFi access points with known positions serve as anchor points.
- Implemented Software
  The implemented software runs on the mobile nodes and is responsible for RSS value collection, PDR and the calculation and tracking of the mobile nodes positions.

4.1 Hardware

4.1.1 Mobile Nodes

Two Android-based cell phones were used in the following experiments as mobile nodes: HTC One and a Sony Xperia Z1 Compact. Details are listed in table 4.1.

TABLE 4.1: Mobile node system description

	HTC1	**Z1C**
Model	HTC One	Sony Xperia Z1 Compact
OS	Android 5.0.1 (API-level 21)	Android 5.1.1 (API-level 22)
CPU	Qualcomm Snapdragon 600 Quad-Core 1.7 GHz	Qualcomm Snapdragon 800 Quad-Core 2.2 GHz
RAM	2 GB	2 GB
Accelerometer (resolution)	BOSCH BMA250 3-axis Accelerometer v1 (0.038307227 m/s^2)	Bosch BMA2X2 Accelerometer/ Temperature/Double-tap v1, (0.07661438 m/s^2)
Magnetometer (resolution)	Asahi Kasei Microdevices AK8963 3-axis Magnetic field sensor v1 (0.06 μT)	Asahi Kasei Microdevices AK8963 Magnetometer v1 (0.14953613 μT)
Note	WiFi scans could be limited to the 2.4 GHz band only, therefore roughly doubling the sampling rate.	

Before each experiment, both magnetometers were calibrated using the compass calibration function of the google maps app. The HTC1 magnetometer precision reached “medium”, while the Z1C precision could not be improved above “low”.

4.1.2 Anchor Nodes

The anchor nodes consist of 5 commercial WiFi access points (D-Link D-625 and DAP-2553) which were distributed over the experimental area (figure 5.2, red squares).

4.2 Mobile Node Software

The algorithm was implemented as an Android-based app to run on the mobile nodes. The structure of the localization core is shown in figure 4.2.
The localization module is responsible for the real-time localization, consisting of the following sub-modules:

- Compass
  This module calculates the current heading using magnetometer and accelerometer data. Due to the low precision of the sensor data and therefore the resulting moving direction, all readings during one step are averaged.
  The moving direction is corrected by the approximate angle of the Institute of Computer Science building to yield the heading in the map coordinate system.
  The compass module is based on a sample compass application [17].

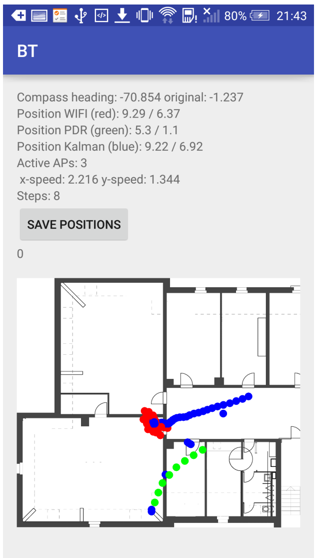
- Step Detector
  Based on accelerometer input, steps are detected.
  Since version 4.4, Android provides a built-in step detector. Since the HTC1 mobile node does not support this type of sensor, step detection was implemented using the available accelerometer sensor based on a project published on GitHub [37].
  After each step, the speed of the mobile node is calculated (see section 2.4.1).
  The speed is averaged over all steps between the WiFi measurements.

- WiFi
  The WiFi module collects the current RSS values of all access points and updates the list of currently active access points in the localization module. A moving average of two values is used to minimize the influence of measurement errors.

- Map
  The map module displays the map of the experimental area and draws the following positions:

  - Kalman filter-derived position (Drawn in blue)
    This position is derived from RSS readings and PDR information, fused by the Kalman filter algorithm.
  - WiFi-only position (red)
    The position as determined only by RSS readings.
  - PDR-only (green)
    This position is only predicted by PDR and not corrected by RSS results or the Kalman filter. Due to error accumulations, it is not reliable.

A screenshot of the application is shown in figure 4.1.



Compass heading: -70.854 original: -1.237
Position WIFI (red): 9.29 / 6.37
Position PDR (green): 5.3 / 1.1
Position Kalman (blue): 9.22 / 6.92
Active APs: 3
x-speed: 2.216 y-speed: 1.344
Steps: 8

0

FIGURE 4.1: The screenshot shows the implemented application in localization mode.

Furthermore, the application contains modules to define anchor points and for performing the calibration.

4.3 Evaluation Scenario

The ILS evaluation was performed on the third floor of the building of the Institute of Computer Science at the University of Bern (Neubrückstrasse 10, 3012 Bern). The office-like area with 288 m^2 size shown in figure 5.1 covered 7 rooms, thereof 2 offices and 2 seminar rooms and a server room.

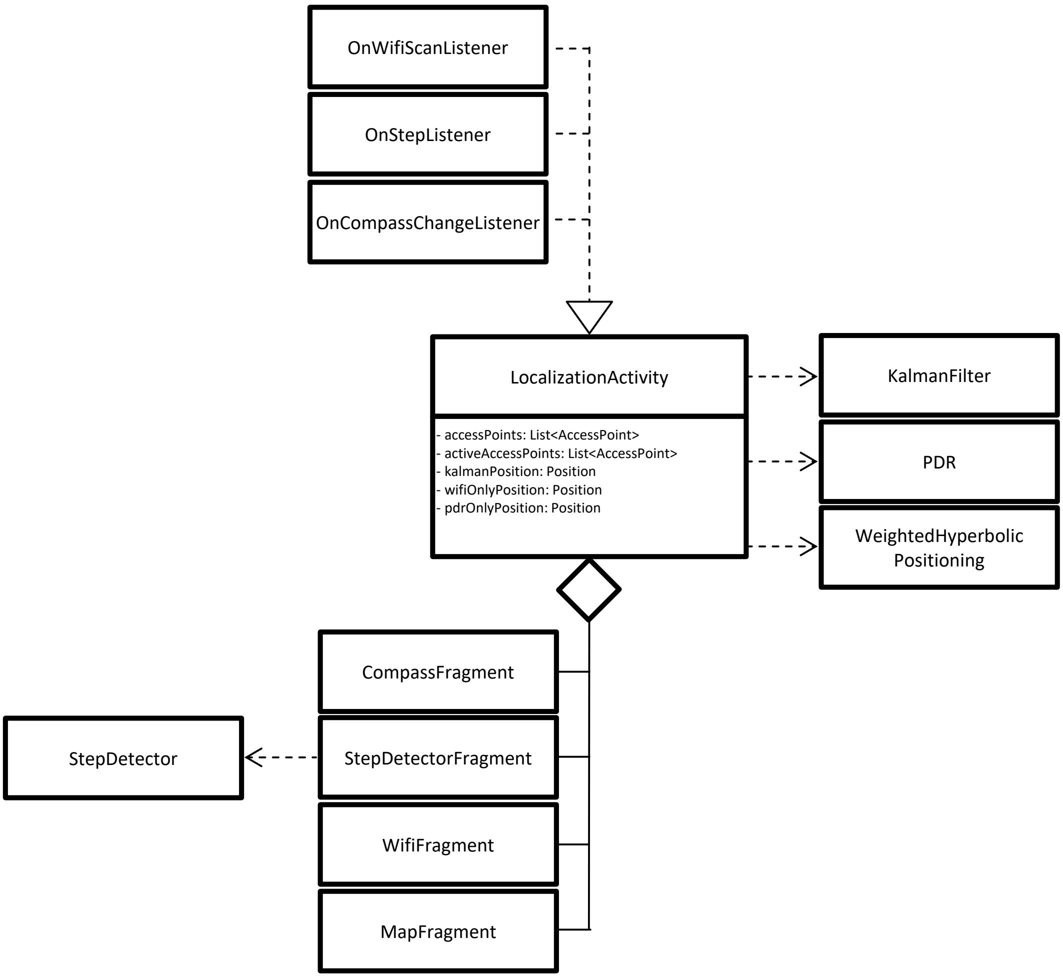

OnWifiScanListener
OnStepListener
OnCompassChangeListener
LocalizationActivity
- accessPoints: List<AccessPoint>
- activeAccessPoints: List<AccessPoint>
- kalmanPosition: Position
- wifiOnlyPosition: Position
- pdrOnlyPosition: Position
KalmanFilter
PDR
WeightedHyperbolic Positioning
CompassFragment
StepDetector
StepDetectorFragment
WifiFragment
MapFragment

FIGURE 4.2: The simplified UML diagram shows only the part related to the real-time localization.

5 Performance Evaluation

This chapter presents the performance evaluation of the implemented system: In section 5.1 the evaluation setup is shown, sections 5.1.1 and 5.1.2 present the system calibration and its verification. Sections 5.2 and 5.3 present the results of the localization evaluation and system resource usage.

5.1 Experiment Setup

To evaluate the ILS localization, an evaluation trajectory shown in figure 5.1 was followed. The mobile nodes were always held identically oriented: In parallel with the ground and top of the mobile node into the direction of walking. This is assumed by the PDR-module. If held otherwise, the PDR-localization would be led astray.
At 18 pre-defined checking points, the current position as defined by the ILS was recorded by the application. To assess the effect of the Kalman filter, the positions provided by the WiFi RSS-based and PDR-based approaches were recorded as well (labelled " WiFi-only" and "PDR-only").

5.1.1 Calibration

To establish the ranging parameters, a calibration phase was necessary before performing the evaluation experiments. The calibration phase is described in more detail in section 3.1.
To assess the validity of the ranging parameters, the resulting α_i and β_i were used to calculate the distances from each calibration point to each access point according to 2.2.1. Resulting distances were compared with the real distances, results thereof are shown in figure 5.3. The mean errors of 1.9 - 2 m seem acceptably low. The maximal errors of 10.3m and 15.1m on the other hand are substantial compared to the experimental area size of 16x18m. In both cases, the measured RSS values were lower than would be expected for this distance. Therefore, the distances were overestimated. This may be explained by the fact that the real distances between AP's and calibration points were large, with several walls in between. Therefore, fading effects are likely to have a stronger influence compared to more closely located AP's. Since the trilateration algorithm uses a distance-depending weighting, the influence of AP's that are further away and therefore probably less accurately estimated distances may not negatively influence the performance of the WiFi-positioning.

5.1.2 Temporal Calibration Data Stability

Validity of the calibration parameters was tested 4 and 6 months respectively after the calibration by repeating the RSS level measurement at the checking points used for localization testing. The distances to the access points were calculated according to the nonlinear regression model, using the parameters defined in the calibration phase (see section 2.2.1).
The ranging errors of this verification set are presented in figure 5.3. They are in the

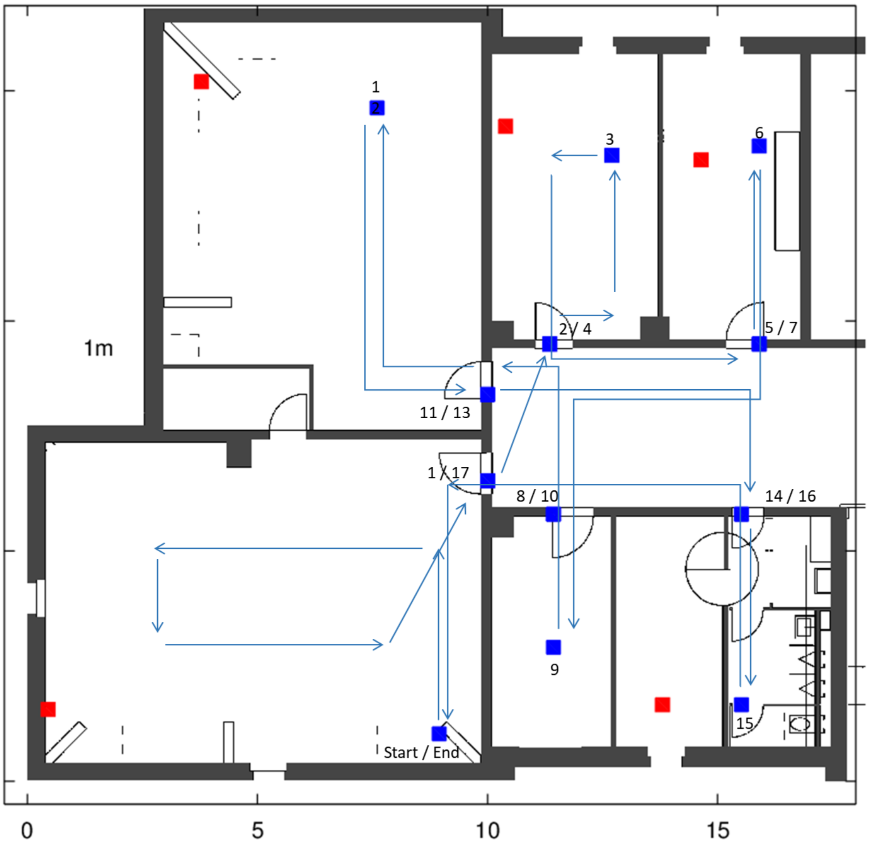


FIGURE 5.1: The trajectory to test the localization implementation is shown in light blue with arrows indicating the direction of movement. Blue squares: Checking points with the corresponding numbers. Red squares: Access points

TABLE 5.1: Ranging Parameters established during the calibration phase. R^2: Coefficient of determination of non-linear fit.

	HTC1			Z1C		
Access Point	α	β	R^2	α	β	R^2
ap1	0.91	-0.043	0.6127	0.0879	-0.076	0.7819
ap2	0.7906	-0.045	0.6143	0.2564	-0.061	0.6479
ap3	0.3042	-0.056	0.7576	0.0741	-0.076	0.7721
ap4	0.2215	-0.061	0.7911	0.0534	-0.081	0.8205
ap5	0.3358	-0.058	0.6382	0.1093	-0.07	0.7443

same range as those of the calibration set.

Therefore, it seems possible to use once acquired calibration parameters for a longer time span. Of course, this requires that the positions of the access points are fixed and the interior fitting remains comparable, otherwise the ranging error may increase.

Since the access point ap3 was moved (to the next room to the right) between the calibration and this verification, the calculated distances to this access point are less accurate.

To get an impression of the device-specificity of the calibration parameters, this verification data set was analysed using the calibration parameters of the other mobile

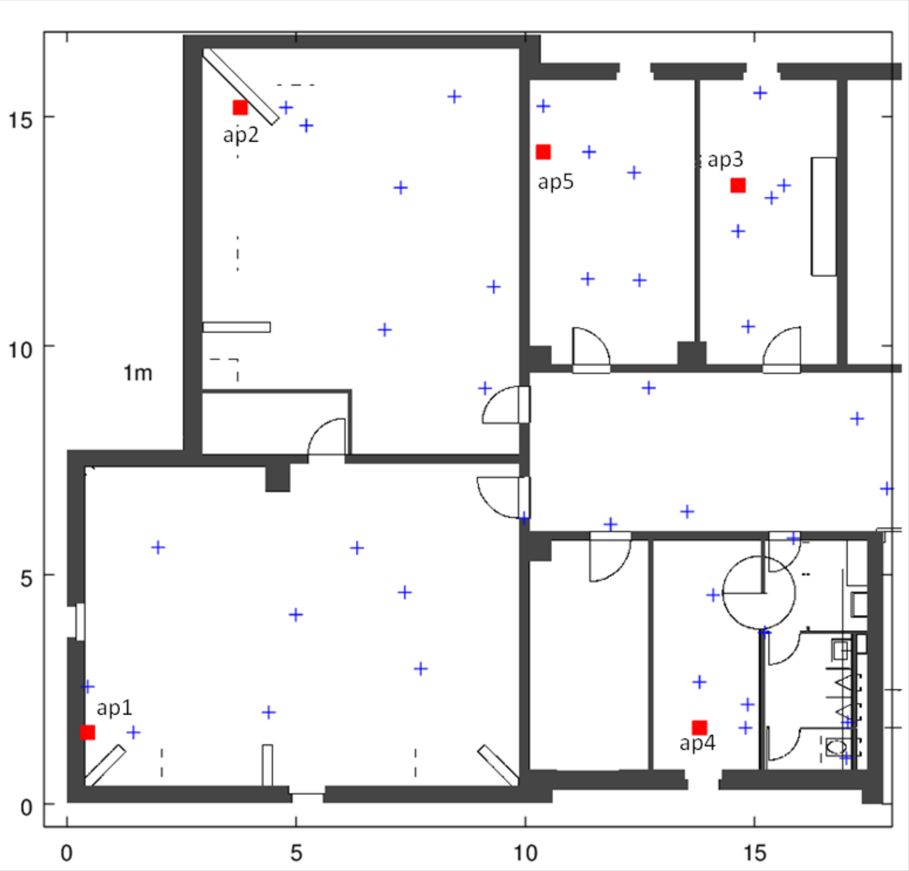


FIGURE 5.2: Map of the evaluation area, showing the positions of the anchor nodes (red squares) and calibration points (blue crosses).

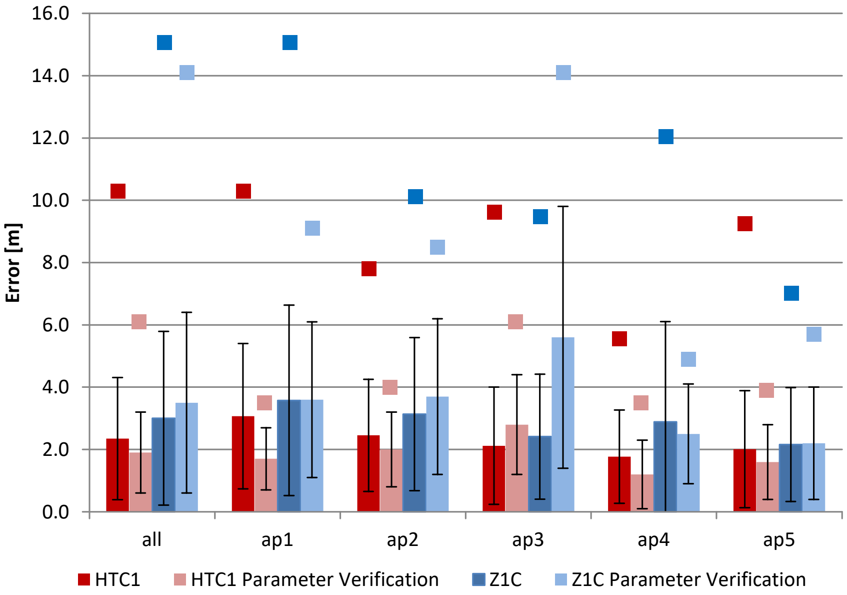


FIGURE 5.3: Ranging errors separated by access point and overall mean. Bars represent standard deviations and single points the maximal errors. Errors of the initial calibration are shown in dark blue and red, errors of the verification data set in light colors.

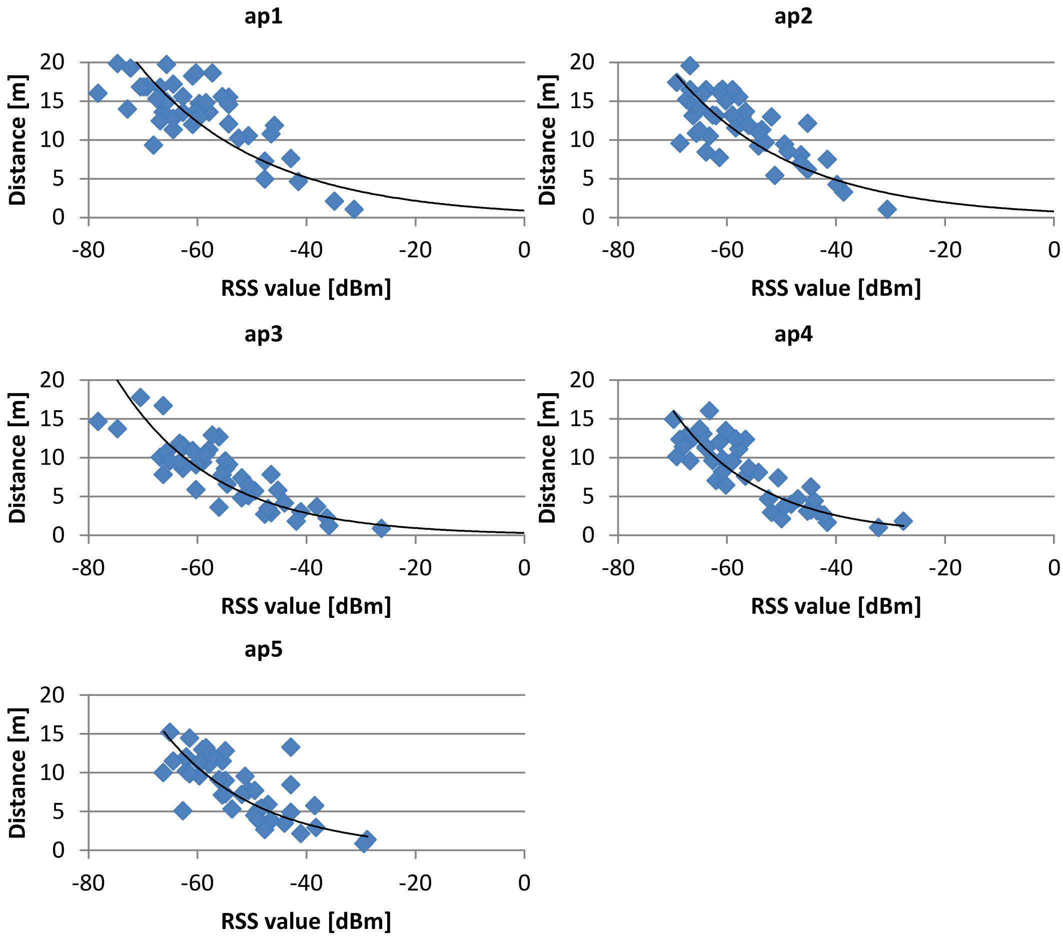


FIGURE 5.4: Exemplary, the calibration data set of the HTC1 mobile node. The non-linear fit model is shown in black.

node. The resulting mean ranging errors were 2.7m and 3.3m for the HTC1 and Z1C respectively. These mean ranging errors are similar to those calculated using the correct calibration parameters. This suggests that the calibration parameters are not strictly device-specific.

5.2 Localization

Evaluation data was exported and analysed on a PC using Microsoft Excel 2013. The measured positions were compared to the actual positions of the checking point. The error at point i, $error_i$ was calculated as the distance between the actual checking point position (x_{i_a}/y_{i_a}) and the measured position (x_{i_m}/y_{i_m}):

$$error_i = \sqrt{(x_{i_a} - x_{i_m})^2 + (y_{i_a} - y_{i_m})^2} \tag{5.1}$$

The covariance matrices used in Kalman filtering contain the standard deviations σ_p, σ_v and σ_a. Their values had to be determined experimentally. The values published by Tarrío and colleagues (“Tarrío parameters”) were used to initially populate the matrices [48]. Other additional sets were tested with the HTC1 mobile node:

- "Tarrío parameters"
  $\sigma_p = 2m, \sigma_v = 0.2m/s, \sigma_a = 0.2m/s^2$
- "PDR-focused"
  $\sigma_p = 2m, \sigma_v = 0.1m/s, \sigma_a = 0.1m/s^2$
- "WiFi-focused'
  $\sigma_p = 1m, \sigma_v = 0.2m/s, \sigma_a = 0.2m/s^2$

The resulting errors of all localization experiments are shown in table 5.2. During the first runs using the Z1C, wrong ranging parameters were used (first part of table 5.2). Therefore, the WiFi-only localization shows pretty high average errors. These were strongly reduced when using appropriate ranging parameters. These runs were excluded when calculating the overall average errors and 90% accuracy values.
For the HTC1 mobile node, several sets of covariance parameters were tested, because the initial set ("Tarrío parameters") did not yield satisfactory results. WiFi- or PDR-focused (table 5.2, bottom) refer to covariance parameter sets that favour the respective type of measurement. With WiFi-focused parameters, the localization performance of the Kalman filter is at least similar to the WiFi-based approach.
The overall localization error of the Kalman filter-based system is 3.3m for the Z1C and 3.2m for the HTC1 respectively.
90% accuracy for both mobile nodes is 4.1m with Kalman filtering, which outperforms the WiFi-only approach by 25% (Z1C) and the PDR-only approach by 59% (Z1C) and 64% (HTC1) respectively. For the HTC1 mobile node, the 90% accuracy for the WiFi-only approach was identical with the Kalman filter accuracy.

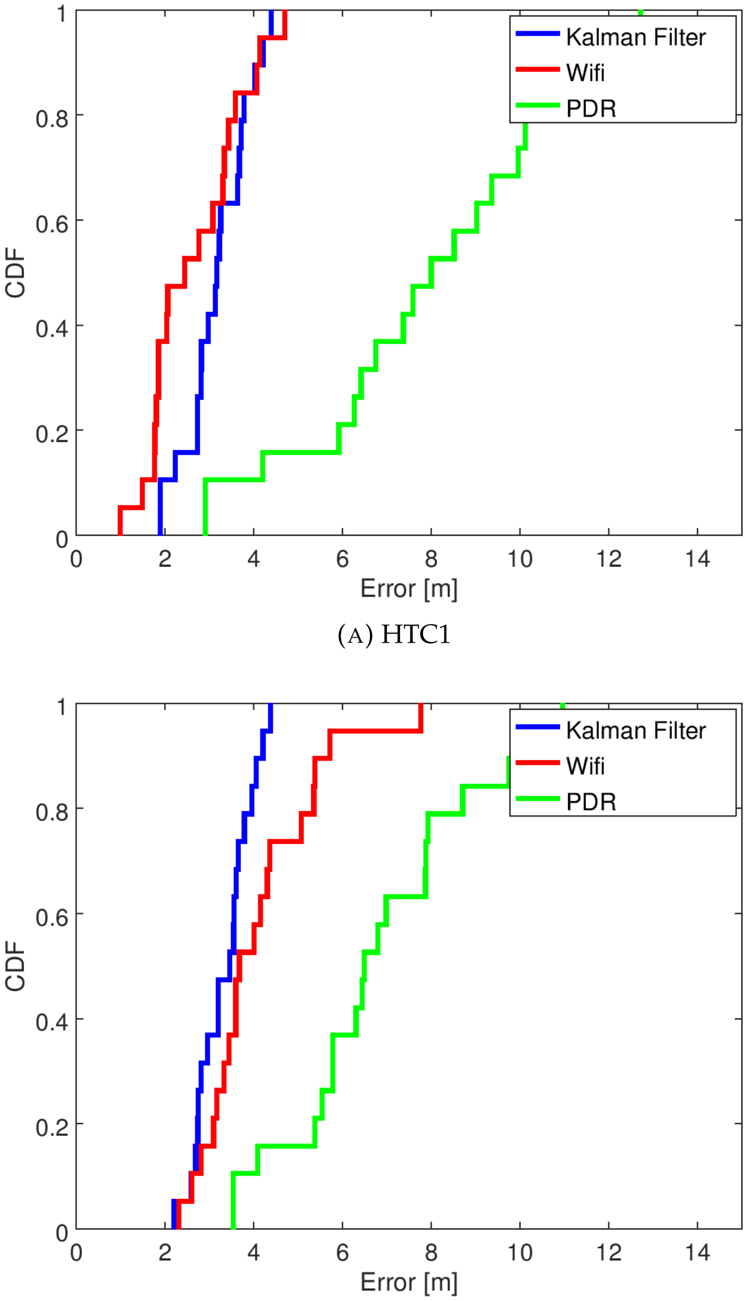
Figure 5.5 shows the cumulative distributed functions (CDF) of the averaged errors for each mobile node. Tested using the Z1C mobile node, the Kalman filter approach was the most accurate, outperforming both the WiFi-only and the PDR-only approaches. Using the second mobile node, Kalman filter approach achieved a similar or slightly lower accuracy, but still outperformed the PDR-only approach.

Showing the mean errors at each checking point in figure 5.6, the error accumulation of the PDR localization is clearly shown. The errors for WiFi only and Kalman filter localization remain stable over the evaluation trajectory.
To further enhance the effect of the Kalman filter, the covariance matrix parameters reflecting the precision of the signal strength and the inertial unit measurements should be further fine-tuned. As shown in table 5.2, HTC1 part, selecting different standard deviation parameters can improve the performance of the Kalman filter.

TABLE 5.2: Average Error and standard deviation (SD) of each run and the mean of all runs with the same set of parameters.
Top: Results for Z1 Compact. Bottom: HTC One mobile node.

Z1C	Average Error per Run			SD of error per run		
	Kalman	WiFi	PDR	Kalman	WiFi	PDR
bad calibration parameters	5.1	6.6	40.5	2.2	2.9	18.7
	6.9	7	19	2.7	3.1	7
	5.8	6.3	11.9	3.1	3.7	4.9
	5.1	6.7	13	2.3	3	3.8
average	5.7	6.7	21.1	0.7	0.3	11.5
good calibration parameters	4.1	5	3.7	2.3	3.5	1.9
	3.3	3.6	11.1	1.3	1.4	4.2
	3.5	4.1	3.2	1.4	2	1.2
	3.2	4.1	4.8	1.4	1.9	2.1
	2.5	3.8	12.4	0.8	3.3	4.9
average	3.3	4.1	7	0.5	0.5	3.9

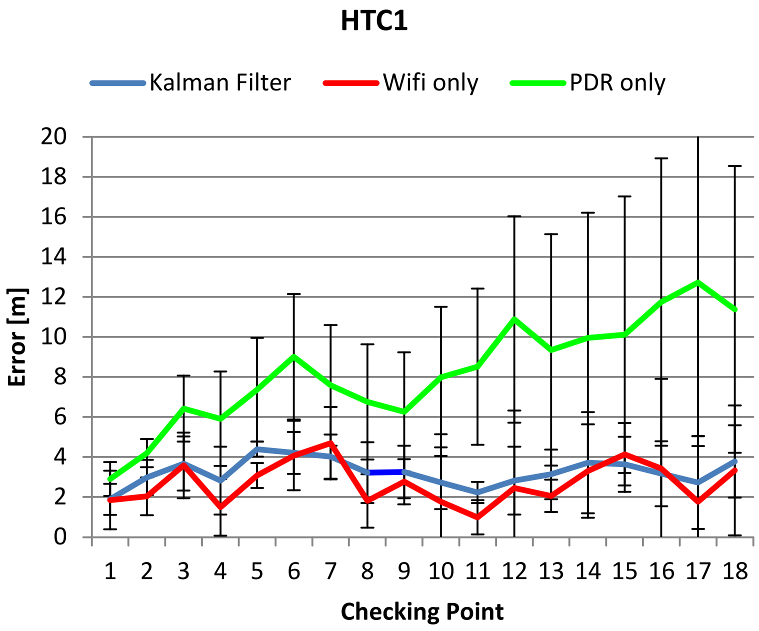
HTC1	Average Error per Run			SD of error per run		
	Kalman	WiFi	PDR	Kalman	WiFi	PDR
“Tarrío parameters”	2.3	2.6	7.9	0.9	1.3	4.2
	2.5	2.7	11.9	0.9	1.5	5.1
	2.7	2.5	14.8	1.4	1.4	4.1
	4.7	4.4	5.6	4.3	3.6	1.8
	1.8	2	5	0.9	1	1.6
	3.1	2.4	5.3	1.1	1.3	1.1
	4.4	3.3	7.6	1.7	1.7	4.8
	4	3.3	4.4	1.9	1.7	1.6
average	3.2	2.9	7.8	1	0.7	3.5
“PDR-focused”	3.4	2.3	9.2	1.8	1.5	2.5
	3.8	2.3	15.3	1.4	1.1	7.8
	3.1	2.1	7.2	1.9	1.1	4.5
	3.7	2.5	9.7	1.7	1	3.7
	3.4	2.7	4.1	2	1.1	1.3
average	3.5	2.4	9.1	0.2	0.2	3.7
“WiFi-focused”	2.2	2.2	11.5	1.6	1.6	2.2
	3.6	3.4	4.7	1.3	1.5	2.1
average	2.9	2.8	8.1	0.7	0.6	3.4



(A) HTC1

(B) Z1C

FIGURE 5.5: Cumulative Distribution Function of Errors. Average of all (HTC1) or only including experiment with correct calibration parameters (Z1C).



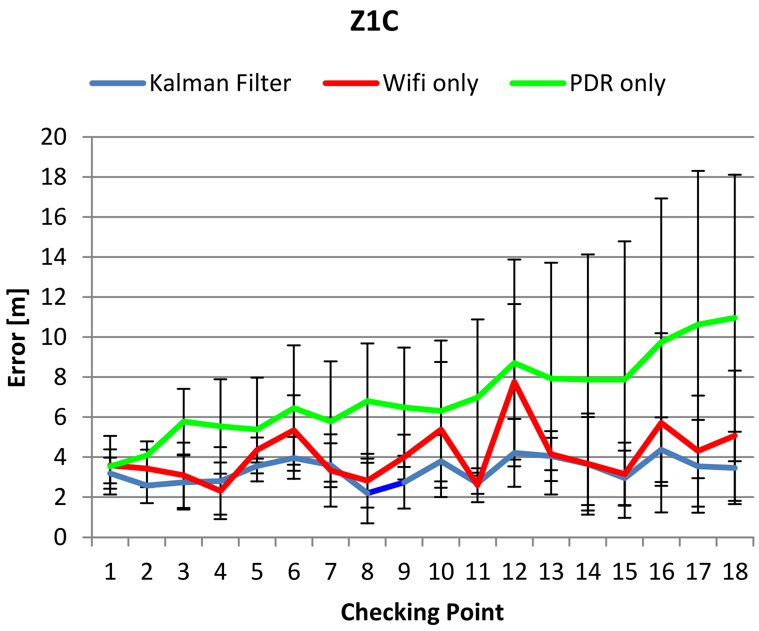


FIGURE 5.6: Average error per checking point.
Top: HTC One, bottom: Z1 Compact. All test runs were used to calculate the average error per checking point. Vertical bars represent ± standard deviation.

5.3 Computational Resources

While accuracy is the most important metric for an indoor positioning system, usage of system resources may be important as well, mainly CPU usage and - most important and linked to CPU usage - battery consumption.
Therefore, both additional metrics were analysed and compared to a particle filter-based localization application with 1200 particles [9]. CPU usage was logged using the freely available tool AnotherMonitor for 5 minutes per application [3].
To monitor battery consumption, Battery Historian was used to analyse android bug report files, generated by the Android Debugging Bridge. This report includes data on battery consumption [16]. Both applications were run for 30 minutes to gather data.

As presented in figure 5.7, the Kalman filter implementation shows a CPU usage of 6% on average, while the particle filter application used 53% of the CPU. In both cases, running background tasks were identical and the CPU usage thereof is included in the resulting CPU usages of the localization applications.
Considering energy consumption – finally one of the most interesting parameters regarding mobile technology – the results are similar. While the usage of a particle filter leads to an averaged energy consumption rate of 2.1W, the running Kalman filter application leads to a consumption rate less than 50% of it (1W). The difference is not as large as for the CPU usage, since WiFi and especially the screen use probably the same amount of energy for both filter types.

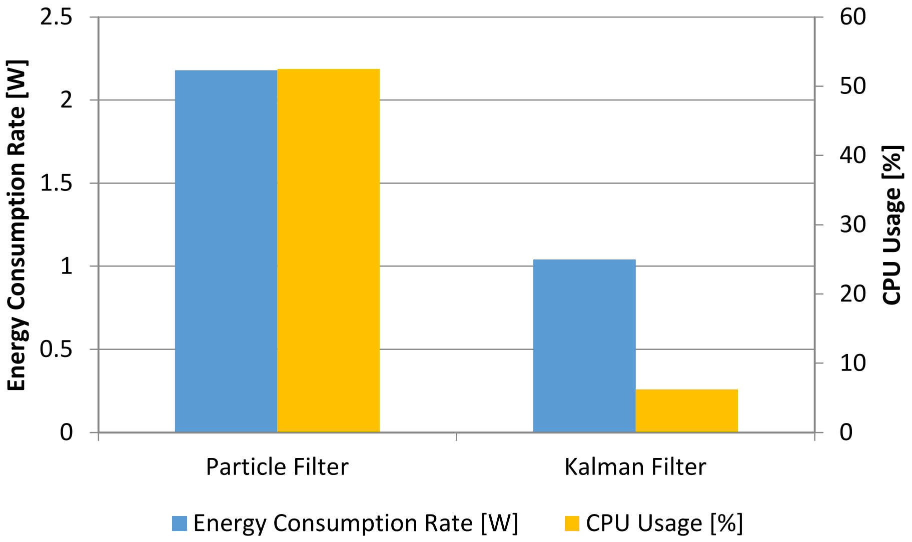


FIGURE 5.7: Usage of mobile node system resources during localization, by filter type. CPU Usage: Mean of 5min measurement (1 value per second).
Energy Consumption: Mean of 8 (Kalman Filter) and 14 (Particle Filter) values respectively.

6 Conclusions

An Android application based on Kalman filter and using WiFi signal strength readings and inertial measurement unit information for localization could be implemented successfully. Furthermore, the system performance was assessed. The resulting average errors and 90% accuracy for the Kalman filter implementation are 2.9m and 4.1m for the HTC1 and 3.3m and 4.1m for the Z1C respectively. As figure 5.5 shows, data fusion by the Kalman filter did only improve the overall accuracy for the Z1C mobile node, but not for the HTC1. In the first case, the Kalman filter reduced the average localization error by roughly 1m.

This finding was unexpected, since the use of a Kalman filter should reduce the effect of noisy measurements and lead to improved overall localization performance. The lack of improvement may probably be explained by the rather high accuracy of the WiFi-only localization of the HTC1 mobile node itself, compared to the PDR-based localization approach. This difference may lead to less accurate results, since the inaccurate PDR-based localization negatively influences the localization.

This assumption is supported by the fact that the Z1C WiFi-localization is less accurate and the Kalman filter had a beneficial effect. The lower accuracy is probably due to the lower WiFi scan frequency and therefore less WiFi position measurements. This introduced a higher potential lag between the last RSS measurement and the position capturing at the checking point, leading to a larger difference between measured and actual position. In contrast to the HTC1 mobile node, the Z1C node could not limit the WiFi scans to the 2.4 GHz band. Not scanning the 5 GHz band approximately doubled the scanning rate for the HTC1, probably improving the accuracy of the WiFi-based localization.

A very similar localization system, also based on WiFi RSS ranging, PDR and a Kalman filter, was implemented by Tarrío and colleagues [48]. They reported a localization accuracy with a mean error of 2.3m with Kalman filter (2.9m for WiFi only, 2.8m for WiFi + PDR). These average error values are comparable to those presented in this thesis. Interestingly, the smaller deployment area with $100m^2$ and the higher number of anchor nodes (n=9) did not lead to an improved WiFi-only localization compared to this work. It is possible that either the high number of calibration points used in this work, or the different ranging algorithm (log-normal channel model versus non-linear regression model) may explain this point.

Using a particle filter to fuse WiFi-signal strength-based positioning information with PDR and map information, José Carrera et al. achieved a significantly more accurate localization with a mean error of 1 to 1.6m, depending on the scenario and settings used [9]. PDR-only accuracy was comparable with 8.6m and 13.7m.

Therefore, the accuracy achieved in this work is in an expected range, but localization systems based on other technologies are significantly more accurate. Therefore, the suitability of a Kalman filter-based localization system depends on the accuracy requirements.

One major advantage of a Kalman filter - and disadvantage of a particle filter - is the low calculation demand and therefore energy consumption. When comparing the localization system presented this thesis and the particle-filter based system by

J.Carrera [9] on the same cell phone, the particle filter implementation uses almost 10-fold more CPU-time. This leads to a 2-fold battery consumption rate. Therefore, if the mobile node to implement a localization system is limited in CPU power and / or battery lifetime, a Kalman filter may be favoured, even if the localization performance is lower.

6.1 Future Work

This work presents a basic implementation of the proposed localization system. Several improvements could be implemented:

- Step Length
  Currently, a fixed step length is assumed. While this works reasonably well if one person is using the system in a pre-defined way, this assumption does not hold true for other people and situations. Therefore, by using accelerometer data, the actual step length would need to be calculated (one method is presented in the work by Tarrío [48]).

- Mobile node orientation
  The PDR-part of the system depends on the mobile node being directed in the direction of movement. Depending on the type of application, this limitation would need to be eliminated.

- Calibration Phase
  A further limitation of this system is the need for calibration data. Although the error difference of ranging is not dramatically different if a set of calibration parameters from another mobile node is used, they do differ for different cell phones. Therefore, calibration parameters should be determined for each phone type or at least family to get the highest accuracy, which is not feasible. To reduce this problem, using some form of relative RSS readings may be of advantage, as used in the "Freeloc" system with a fingerprinting approach [31].

- Error Covariance Matrices
  To possibly improve the system accuracy, the standard deviation parameters of the process and measurement covariance matrices should be further fine-tuned.

Bibliography

[1] Estefanía Aguilar-Moreno, Raúl Montoliu-Colás, and Joaquín Torres-Sospedra. "Indoor positioning technologies for academic libraries: towards the smart library". In: *Profesional De La Informacion* (2016), pp. 295–302.

[2] Sardar Ansari et al. "An extended Kalman filter with inequality constraints for real-time detection of intradialytic hypotension". In: *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (2017).

[3] Retondo Antonio. *AnotherMonitor*. [Online; accessed 05.06.2017]. 2015. URL: https://github.com/AntonioRedondo/AnotherMonitor.

[4] Yan Bai and Xiaochun Lu. "Research on UWB indoor positioning based on TDOA technique". In: *9th International Conference on Electronic Measurements and Instruments* (2009), pp. 1167–1170.

[5] Stéphane Beauregard and Harald Haas. "Pedestrian Dead Reckoning: A Basis for Personal Posiitoning". In: *Proceedings of the 3rd Workshop on Positioning, Navigation and Communication* (2006).

[6] U. Biader Ceipidor et al. "SeSaMoNet: an RFID-based economically viable navigation system for the visually impaired". In: *International Journal of RF Technologies: Research and Applications* (2009).

[7] Maged N Kamel Boulos and Geoff Berry. "Real-time Locating Systems (RTLS) in Healthcare: a condensed primer". In: *International Journal of Health Geographics* (2012).

[8] Ramon F. Brena et al. "Evolution of Indoor Positioning Technologies: A Survey". In: *Journal of Sensors* (2017).

[9] José Luis Carrera et al. "A Real-time Indoor Tracking System in Smartphones". In: *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (2016).

[10] Zhenghua Chen et al. "Fusion of WiFi, Smartphone Sensors and Landmarks Using the Kalman Filter for Indoor Localization". In: *Sensors* 15 (2015).

[11] European Comission. *SESAMONET - improved mobility of the visually impaired*. [Online; accessed 18.08.2017]. 2007. URL: https://ec.europa.eu/jrc/en/page/sesamonet-improved-mobility-visually-impaired-9350.

[12] Ramsey Faragher. "Understanding the Basis of the Kalman Filter Via s Simple and Intuitive Derivation". In: *IEEE Signal Processing Magazine* (2012), pp. 128–132.

[13] Silke Feldmann et al. "An Indoor Bluetooth-Based Positioning System: Concept, Implementation and Experimental Evaluation". In: *Proceedings of the International Conference on Wireless Networks* (2003).

[14] Andreas Fink et al. "RSSI-based Indoor Positioning using Diversity and Inertial Navigation". In: *International Conference on Indoor Positioning and Indoor Navigation* (2010).

[15] Carlos E. Galván-Tejada et al. "Infrastructure-Less Indoor Localization Using the Microphone, Magnetometer and Light Sensor of a Smartphone". In: *Sensors* 15 (2015), pp. 20355–20372.

[16] Google. *Battery Historian*. [Online; accessed 05.06.2017]. 2015. URL: https://github.com/google/battery-historian.

[17] Fernando Greenyway. *Using orientation sensors: Simple Compass sample*. [Online; accessed 14.05.2016]. 2011. URL: http://www.codingforandroid.com/2011/01/using-orientation-sensors-simple.html.

[18] Paul D. Groves. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Boston: Artech House, 2008.

[19] Yanying Gu, Anthony Lo, and Ignas Niemegeers. "A Survey of Indoor Positioning Systems for Wireless Personal Networks". In: *IEEE Communications Surveys & Tutorials* (2009).

[20] Dominik Gusenbauer, Carsten Isert, and Jens Krösche. "Self-Contained Indoor Positioning on off-the-shelf Mobile Devices". In: *International Conference on Indoor Positioning and Indoor Navigation* (2010).

[21] G. Gustafsson and F. Gunnarsson. "Mobile Positioning Using Wireless Networks". In: *IEEE Signal Processing Magazine* (2005).

[22] M.F.S van der Ham et al. "Real Time Localization of Assets in Hospitals Using Quuppa Indoor Positioning Technology". In: *Annals of the Photogammetry, Remote Sensing and Spatial Information Sciences, International Conference on Smart Data and Smart Cities* (2016).

[23] Tom van Haute et al. "Performance analysis of multiple Indoor Positioning Systems in a healthcare environment". In: *International Journal of Health Geographics* (2015).

[24] Anna Heinemann et al. "RSSI-Based Real-Time Indoor Positioning Using ZigBee Technology for Security Applications". In: *Multimedia Communications, Services and Security*. 2014.

[25] Feng Hong et al. "WaP: Indoor Localization and Tracking Using WiFi-Assisted Particle Filter". In: *39th Annual IEEE Conference on Local Computer Networks* (2014).

[26] Tien-Chi Huang et al. "Get lost in the library?: An innovative application of augmented reality and indoor positioning technologies". In: *The Electronic Library* (2016).

[27] Infsoft. *Indoor Navigation & Services in Airports*. [Online; accessed 04.06.2017]. URL: https://www.infsoft.com/industries/airports/features.

[28] Simon J. Julier and Jeffrey K. Uhlmann. "Unscented filtering and nonlinear estimation". In: *Proceedings of the IEEE* 92 (2004).

[29] Rudolf E. Kálmán. "A New Approach to Linear Filtering and Predition Problems". In: *Journal of Basic Engineering* (1960), pp. 35–45.

[30] Moritz Kessel, Martin Werner, and Claudia Linnhoff-Popien. "Compass and WLAN Integration for Indoor Tracking on Mobile Phones". In: *UBICOMM 2012: The Sixth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies* (2012).

[31] Wooseong Kim et al. "Crowdsource Based Indoor Localization by Uncalibrated Heterogeneous Wi-Fi Devices". In: *Mobile Information Systems* (2016).

[32] Johann Larsson. “Estimation of dialysis treatment efficiency by means of system identification”. In: *Master Thesis, Lund University* (2014).

[33] Tommer Leyvand et al. “Kinect Identity: Technology and Experience”. In: *Entertainment Computing* (2011).

[34] Xinrong Li. “RSS-Based Location Estimation with Unknown Pathloss Model”. In: *IEEE Transactions on WIreless Communications* 5 (2006).

[35] Zan Li, Torsten Braun, and Desislava C. Dimitrova. “A Passive WiFi Source Localization System based on Fine-grained Power-based Trilateration”. In: *World of Wireless, Mobile and Multimedia Networks* (2015).

[36] Perpaolo Loreti, Paolo Sperandio, and Massimo Baldasseroni. “A Multi-Technology Indoor Positioning Service to Enable New Location-aware Applications”. In: *2012 IEEE First AESS European Conference on Satellite Telecommunications* (2012).

[37] lufo816. *Pedometer*. [Online; accessed 04.06.2017]. 2015. URL: `https://github.com/lufo816/Pedometer`.

[38] Rainer Mautz. “Indoor Positioning Technologies”. In: *Habilitation Thesis, ETH Zürich* (2012).

[39] NATO. *Theory and Applications of Kalman Filtering*. London, 1970.

[40] L.M. Ni et al. “LANDMARC: indoor location sensing using active RFID”. In: *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications* (2003).

[41] Lei Niu. “A Survey of Wireless Indoor Positioning Technology for Fire Emergency Routing”. In: *8th International Symposium of the Digital Earth* (2014).

[42] Irfan Oksar. “A Bluethooth Signal Strength Based Indoor Localization Method”. In: *21th International Conference on Systems, Signals and Image Processing* (2014).

[43] G Prasad et al. “Plant-wide predictive control for a thermal power plant based on a physical plant model”. In: *IEE Proceedings - Control Theory and Applications* 147 (2000).

[44] T. Rappaport. *Wireless communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.

[45] David Salmond and Neil Gordon. “An introduction to particle filters”. In: (2005), pp. 128–132. URL: `http://appliedmaths.sun.ac.za/~herbst/MachineLearning/ExtraNotes/ParticleFilters.pdf`.

[46] Peter Stephan et al. “Evaluation of Indoor Positioning Technologies under industrial application conditions in the SmartFactoryKL based on EN ISO 9283”. In: *IFAC Proceedings Volumes* 42 (2009), pp. 870–875.

[47] P Tarrío, M. Bernardos A., and R. Casar J. “Weighted Least Squares Techniques for Improved Received Signal Strength Based Localization”. In: *Sensors* (2011), pp. 8569–8592.

[48] P. Tarrío, J.A. Besada, and J.R. Casar. “Fusion of RSS and Inertial Measurements for Calibration-Free Indoor Pedestrian Tracking”. In: *16th International Conference on Information Fusion* (2013).

[49] Roy Want et al. “The Active Badge Location System”. In: *ACM Transactions on Information Systems* (1992).

[50] A. Ward, A. Jones, and A. Hopper. “A new Location Technique for the Active Office”. In: *IEEE Personal Communications* 4 (1997).

[51] Ron Weinstein. “RFID: a technical overview and its application to the enterprise”. In: *IT Professional* 7 (2005).

[52] Andrew Wheeler. “Commercial Applications of Wireless Sensor Networks Using ZigBee”. In: *IEEE Communications Magazine* 45 (2007).

[53] Chengshan Xiao and D.J. Hill. “Stability and absence of overflow oscillations for 2-D discrete-time systems”. In: *IEEE Transactions on Signal Processing* 44 (1996).

[54] Weizhi Zhang, Sakib M.I. Chowdhury, and Mohsen Kavehrad. “Asynchronous indoor positioning system based on visible light communications”. In: *Optical Engineering* (2014).

[55] Jason Zhi Liang et al. “Image Based Localization in Indoor Environments”. In: *International Conference on Virtual Systems and Multimedia* (2010).