UNIVERSITY OF BERN

BACHELOR THESIS

Indoor positioning using Raspberry Pi with UWB

Author:
Mischa WENGER

Supervisors:
Jose CARRERA,
Zhongliang ZHAO

Head of Research

PROFESSOR DR. TORSTEN BRAUN

Communication and Distributed Systems
Institute of Computer Science

October 30, 2018

Declaration of Authorship

I, Mischa WENGER, declare that this thesis titled, "Indoor positioning using Raspberry Pi with UWB" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UNIVERSITY OF BERN

Faculty Name
Institute of Computer Science

Bachelor of Science in Computer Science

Indoor positioning using Raspberry Pi with UWB

by Mischa WENGER

Abstract

The number of mobile applications as well as the number of mobile devices is growing continously, which led to more and more interest in the research field of indoor positioning. Many indoor localization solutions rely on common radio signal strength indication or on Inertial Measurement Units (IMU). In this work we present a server-based indoor positioning algorithm based on time of flight measurements of ultra wideband (UWB) signals for ranging, IMUs for move detection and floor plan information. We implemented a particle filter to fuse all these information to achieve high indoor localization performance. We evaluated our system, running on Raspberry Pi devices equipped with Sequitur Pi UWB transmitters, in complex real indoor environments. Moreover, we compared our system to the commercial indoor localization system Sequitur InGPS Lite, distributed by UNISET company. Results show that our algorithm could achieve an average tracking error of $0.45m$ and a 90% accuracy of $0.87m$. Thus, our first prototype can keep up with the Sequitur InGPS Lite system and outperforms previous signal strength implementations, which makes it highly promising for future research.

Contents

List of Figures

List of Tables

List of Abbreviations

ACK	**ACK**nowledgement
AN	**A**nchor **N**ode
API	**A**pplication **P**rogramming **I**nterface
CDS	**C**ommunication **D**istributed **S**ystem
dB	**D**eci**B**ell
GHz	**G**iga**H**ertz
GPIO	**G**eneral **P**urpose **I**nput **O**utput
GPS	**G**lobal **P**ositioning **S**ystem
HDE	**H**euristic **D**rift **E**limination
IMU	**I**nertial **M**easurement **U**nits
IoT	**I**nternet **o**f **T**hings
kbps	**k**ilo **b**it **p**er **s**econd
KNN	**K** **N**earest **N**eighbor
LDPL	**L**og-normal **D**istane **P**ath **L**oss
Mbps	**M**ega **b**it **p**er **s**econd
MCL	**M**onte **C**arlo **L**ocalization
MF	**M**agnetic **F**ield
MHz	**M**ega**H**ertz
ML	**M**achine **L**earning
M2M	**M**achine **2**(to) **M**achine
NLR	**N**on-**L**inear **R**egression
OS	**O**perating **S**ystem
PRF	**P**ulse **R**epetition **F**requency
RTLS	**R**eal **T**ime **L**ocating **S**ystem
RTT	**R**ound **T**rip **T**ime
RSSI	**R**eceived **S**ignal **S**trengh **I**ndication
SDS-TWR	**S**ymmetrical **D**ouble **S**ided - **T**wo **W**ay **R**anging
S.D	**S**tandard **.** **D**eviation
TAG	**TA**r**G**et
TDOA	**T**ime **D**ifference **O**f **A**rrival
ToF	**T**ime **o**f **F**light
TWR	**T**wo **W**ay **R**anging
UDP	**U**ser **D**atagram **P**rotocol
UWB	**U**ltra **W**ide**B**and

Chapter 1

Introduction

In the past few years a new market of mobile gadgets and connected devices, summed up as Internet of Things (IoT), has evolved. In 2017, more than 20 Billion devices were connected to the internet. Forecasts predict already more than 70 Billion in 2025 [23]. This increase in mobile computing has also increased the demand of accurate real-time positioning systems, which led to an active research mainly in indoor positioning system technologies, as there are established solutions for outdoor positioning. In the following we will shortly present our motivation for this work, our contribution and an overview of the remainder of this document.

1.1 Motivation

Location-based applications can be applied in many different indoor contexts, such as entertainment, health, logistic etc. Due to the environmental conditions indoors, with heavy walls armoured with steel and other interferences, additional signal loss is encountered which makes it hard to detect and decode GPS - the established outdoor global positioning system (GPS) - signals [4]. This means that we are forced to use alternative technologies that provide higher accurracy indoors. Although there are many different approaches to do indoor positioning, it can still be considered as an open challenging problem, which made it an attractive and active research field.

Multilateration positioning using radio signals is one of the most widely used indoor localization approach. Radio-based multilateration positioning uses the emitted radio signal power from several transmitters to estimate the position of a target. Most often, WiFi signals are used because they are already omnipresent in indoor environments. However, the radio signal strength is severly affected by environmental conditions, such as furniture, heavy walls etc.

Many devices have various embedded inertial sensors - such as most modern smartphones - or can be equipped with additional sensors. These sensors, e.g. accelerometer, gyroscope and magnetic field sensors, are used to register relative movements of the target. However, relative measurements accumulate errors over time, what leads to extensive long term errors.

Some positioning systems improve the accuracy by using both techniques, absolute radio-based tracking and relative - inertial measurement unit (IMU) based - tracking. By combining these two approaches, it is possible to minimize the positioning errors, as the absolute measures can eliminate long term accumulated errors of the relative system.

Positioning systems can either be client-based or server-based. Most of the positioning systems - especially for smartphones - are client-based, such that the position estimation is done on the target device itself. In a server-based tracking system, the target device sends all recorded data to an external server. The server processes the data and transmits the target position to the client.

An accurate indoor positioning system encounters challenging problems. The noise in low-cost IMUs, as well as the environmental conditions affecting radio-signals will introduce errors in the tracking process. Moreover, many positioning systems are client-based, they run directly on the target device with limited computational power and limited energy source, which adds another difficulty.

1.2 Contributions

In this thesis we present a real-time indoor tracking system for continous positioning and tracking. Our server-based approach provides high accuracy by combining radio, IMU and floorplan information in an enhanced particle filter. We fused ultra wideband radio ranging information with IMU motion detection and with room recognition using ultra wideband and WiFi fingerprinting.

We prototyped our approach on Raspberry Pi devices as anchors and as client. The localization algorithms were implemented in an external, centralized server. Evaluation results show that our system can keep up with commercial tracking systems. It can achieve an average positioning error of $0.49m$ and a standard deviation of $0.24m$.

Our main contributions are:

- We implemented a centralized, server-based real-time indoor localization system
- We fused UWB ranging information, IMU sensor data, floorplan constraints as well as WiFi and UWB room recognition fingerprinting in a particle filter to provide high localization performance.
- We evaluted our server-based system in real complex indoor scenario, where we placed several anchor nodes in a real building and collected data on several indoor trajectories.

1.3 Overview

Our work compounds of five remaining chapters: Chapter 2 provides related work. Chapter 3 presents the theoretical background and chapter 4 highlights our system architecture. In chapter 5 we explain our implementation. The experiment setup and evaluation of our experiments can be found in section 6. Finally the last part concludes the work, where our findings are summarized.

Chapter 2

Related Work

Accurate indoor localization has been examined for a long time. Many different solutions have been developed and presented, using different approaches regarding system architecture and localization method. Within these solutions, mostly client-based architectures can be seen. Moreover, most research focuses on radio-based localization or sensor-based tracking.

In this section we present some related work, grouped by the different types of localization systems.

2.1 Client-based and Server-based Architecture

Localization algorithms can either run on a centralized server or on the client device itself. A server-based localization system can be interesting, as it does not require a specific target device hardware, which makes it fairly scalable. Coordination of the different system components and data processing can be done on the server, such as in [8]. In [5] the system runs on a commodity smartphone, with the benefit of eliminating further communication to an external entity. Such client-based localization systems can often be deployed without any additional hardware, when for example WLAN access points are already available. The authors of [1] use a hybrid of server and client-based architecture - ranges are calculated on the client device and network control as well as the database are managed on a server.

2.2 Fingerprint Localization

Fingerprint Localization is a range-free localization method where radio signals and other measurements that are highly affected by environmental conditions are fingerprinted and stored in a map. The position is estimated by comparing a current fingerprints to the existing map. The authors of [16] provided a room-based ensemble learning technique for localization. The room detection uses averaged coordinate outputs of a k-NN estimator. Whereas in [28] the authors propose a hidden markov model discriminative learning method for indoor localization. Their apporach is a zone recognition algorithm based on magnetic field and WiFi fingerprints brought together with transition probabilities between zones.

2.3 Range-based Localization

In range-based localization systems, range is defined as the propagation distance between the target and anchor nodes (AN). First the propagation distances are calculated, afterwards many different algorithms can be used to find the absolute position of the target. In [27] a received signal strenght indicator is used to estimate the range during the ranging process. A different method to do ranging is used in [7], where the authors calculated distances by the elapsed time between sending and receiving radio messages. Range-based algorithms are often much lighter and computationally less expensive than fingerprinting methods, as the big effort in generating, storing and processing a radio map falls away.

2.4 Pedestrian Dead Reckoning

PDR relies on inertial measurement unit (IMU) readings to find the new position. PDR systems are often not able to calculate absolute positions, but the relative change in position. This leads to an accumulation of errors over time, which are in most systems eliminated by adding a different source of information, such as WiFi signals or floorplan information. Different IMU sensor readings are used to obtain a stride length estimation, a heading direction estimation and step recognition. In [3] gyroscope data is used to determine the heading orientation and accelerometer readings provide the displacement. They defined a method called Heuristic Drift Elimination (HDE) to minimize the accumulated errors, by adding a specialized sensor deployed on the foot of the pedestrian. An other method to find heading direction is used in [12], where the authors use a kind of digital compass by measuring magnetic field energy. Based on accelerometer readings they defined a walking and a running model.

2.5 Hybrid Localization Approaches

The different characteristics of different types of localization methods makes it obvious, that combined localization systems could achieve higher accuracy. In combined approaches, especially relative and absolute measurements are fused to have the advantages of both methods. The authors of [15] used a fingerprinting-based solution in combination with a digital compass. [5] is a particle filter approach that fuses PDR and radio-based ranging, as well as floorplan information into the localization process. Errors in the PDR system are mitigated by the ranging estimations and vice-versa. This makes these approaches highly interesting in the research.

Chapter 3

Background Theory

Localization systems come with various architectures and system designs. Several different methodologies can be applied to estimate the targets current position. In this chapter we distinguish between client and server-based system architectures. We introduce fingerprinting-based and range-based localization techniques, as well as the principles of movement detection. The process of information fusion from different data sources and the radio technology of ultra wideband is explained in the last part of this section.

3.1 Client and Server-based Localization

For many applications the system architecture is very important. The environmental conditions, the underlying hardware and also dogmatic thoughts are taken into account for real applications to determine the system architecture. There are two main types of system architectures - client-based or server-based - that can be distinguished. However, systems using computational resources of client and servers are possible, where a clear distinction is no longer possible.

Client-based architectures have the huge advantage, that no additional server hardware is needed. The client device collects data used for the localization and processes it, to estimate its own position. A client-based localization should be used when standardized target devices with enough computational power and enough energy supply are localized. As all computations are done on the device itself, no further communication to a seperated server is needed. This ensures that no other application can access the position estimation and reduces the communication overhead.

A server-based system however, can be more powerful and computationally complex than client-based systems, as there are no big hardware restrictions. The application itself runs on a centralized server, whereas the client-device is only involved in the data collection process. The requirements of the client-hardware shrink to a minimum, which allows also to use different, non-standardized, devices as target. Especially when position estimations are not used in applications running on the target device, a server-based architecture is beneficial. Often, multiple targets are tracked, which is easily possible with a centralized server approach, as the data of several target devices is directly available on the server and can be further processed.

3.2 Fingerprinting Localization Technique

Fingerprinting Localization, also known as radio map based technique, use a dense positioning of anchor nodes in the indoor area of interest. A set of measurements, often received signal strength information (RSSI), serve as a fingerprint at each location. Measurments are not limited to radio signals, other sources such as magnetic field data can also be used. The more unique these measurements are, the better is the localization accuracy. Fingerprinting often consists of an offline phase to generate the radio map and an online phase to retrieve the position with a given fingerprint.

In the offline phase, a radio map is generated with several fingerprints. The radio map can either consist of different fingerprint measurements at given reference points (landmarks) that are interpolated to the whole area, or of fingerprints measured in predefined zones in the area of interest. After this phase, for every location in the grid, a tupel (location, fingerprint) with a unique fingerprint should be available.

The online phase consists of an observed fingerprint at the targets location, on which the localization algorithm can be applied to associate the fingerprint to similar radio map entries. This association is then used to estimate the targets location. The result can either be a concrete position estimation built on the presence of reference points or it can be a probability of the target being in a recognized zone.

3.3 Range Based Localization

Range based localization systems are depending on an infrastructure in the area of the localization:

- **Target Node (TAG)** which is the device that is localized.
- **Anchor Nodes (AN)** that are placed on carefully chosen points in the building, to encounter the best coverage of the whole area.

The key idea of range based positioning is to measure the distance between TAG and ANs. With the use of these distances, the exact position of the TAG can be evaluated using multilateration or similar mathematical models. Several different approaches are possible to determine the distance to an anchor node, they can be classified into two groups. In the one hand there are algorithms using propagation models, which rely on the reduction in power density of electromagnetic waves propagating through space. In the other hand, algorithms make use of known propagation velocities for radio signals by measuring the time of flight of transmitted waves.

3.3.1 Propagation Models

An electromagnetic wave loses power density when travelling through space. In free space, formula 3.1 explains the relationship between distance and signal strength [24].

$$P_r = P_t(\lambda/4\pi r)^2 \tag{3.1}$$

where P_r is the received signal strength and P_t the transmitted signal strength. λ is the wavelength and r the radius, or in other words the distance from transmitter to receiver. As this formula is restricted to free-space and often in real indoor environments various kind of obstacles are present, several other approximations for the relationship between distance and signal power exist. Many indoor positioning algorithms use received signal strength indication (RSSI) to calculate distances to the anchor nodes. Mainly because RSSI can be applied to almost every type of transmitted signal, thus RSSI uses universal applicable theory. Two approximations are widely used to take care of the non-free-space environments indoors.

A commonly used model for the relationship of distance and RSS is known as Log-normal Distance Path Loss (LDPL), where the path loss in Decibel (dB) is defined as [2]:

$$PL = PL_0 + 10\gamma log_{10}(\frac{d}{d_0}) + X_g \tag{3.2}$$

with γ as the path loss exponent, PL_0 a path loss measurement at reference distance d_0 and X_g a zero-mean Gaussian noise. This generic model can be applied to many different environments. It can even be simplified by defining useful reference distances, like done in [14].

LDPL has shown to be rather inaccurate for indoor environments, which led to a path loss model based on Non-Linear Regression (NLR) as used in [6]. In this approach, the distance to RSS relationship is modelled with the equation:

$$d_i = \alpha_i e^{\beta_i RSS_i}. \tag{3.3}$$

d_i is the distance between the i-th AN and the TAG, RSS_i the measured signal strength at the i-th AN and α_i, β_i specific coefficients obtained in the area of interest. These two coefficients are rather important for the performance of this approach. They are defined due to extensive calculations based on preliminary measuements as described in [14].

3.3.2 Time of Flight Based Models

A totally different technique to get distance estimations are time of flight based models. Their key idea is that we know the travelling velocity of radio waves, which is approximately the speed of light ($2.997\,924\,58 \times 10^8\,\mathrm{m\,s^{-1}}$). When measuring the time between sending and receiving a radio signal, we can easily convert this time of flight (ToF) to a distance estimation. Time of flights are determined by gathering round trip times (RTT) in radio signal communication. For accurate RTT results the hardware of transmitter and receiver, as well as the operating firmware are very important. For the two presented RTT-measuring techniques, the key characteristics are either a quick responding time or extremely well synchronized sender and receiver.

Two way ranging (TWR) is one of these methods to retrieve an accurate round trip time. An illustration of the TWR process can be seen in figure 3.1. When operating in TWR mode, the TAG sends a message to the ANs and registers the exact time of sending. As soon as the message arrives at its destination, the firmware of the AN instantly captures another timestamp. In an acknowledgement (ACK) message, the timestamp of reception and a timestamp of sending the response is transmitted. When this message arrives at the TAG, again a timestamp is registered. With equation 3.4, the time of flight can now be evaluated [22].

To achieve higher accuracy, the communication can be extended with a final message containing two additional timestamps of the requester. This is called symmetrical double-sided two-way ranging (SDS-TWR) [26] and is indicated as optional in Figure 3.1.

Evaluation of time of flight for TWR:

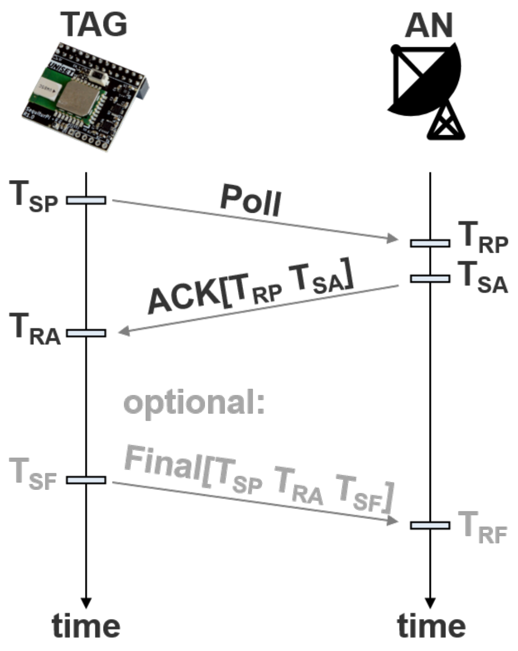

FIGURE 3.1: Illustration of TWR and SDS-TWR communication

$$ToF = [(T_{RA} - T_{SP}) - (T_{SA} - T_{RP})]/2 \tag{3.4}$$

Respectively the time of flight for SDS-TWR:

$$ToF = [(T_{RA} - T_{SP}) - (T_{SA} - T_{RP}) + (T_{RF} - T_{SA}) - (T_{SF} - T_{RA})]/4 \tag{3.5}$$

with the following timestamps, also indicated in figure 3.1: T_{SP} as time of poll-sending, T_{RP} as time of poll-reception, T_{SA} as time of ACK-sending, T_{RA} as time of ACK-reception, T_{SF} as time of Final-sending and T_{RF} as time of Final-reception.

A second approach for RTT determination is called time difference of arrival (TDOA). While TWR does not need further synchronization between the devices, TDOA requires a very precise synchronization of the anchor nodes. This is normally done by specifying a master node per three to five anchors. For bigger scenarios often multiple dedicated masters will send clock synchronizations every once in a while, such that every AN gets at least one sync package. It occurs as well that an anchor holds two differently synchronized times. To evaluate the time of flight, a TAG in range will broadcast a blink message. This blink message will initialize the ranging process. Every AN that receives this blink, will capture a timestamp of the time of arrival (or when holding more than one synctime, capture multiple timestamps). These timestamps are forwarded to the server together with a synch ID and a blink transmitter identity. When a server received at least three timestamps with the same synch ID, it can perform the position and therefore the distance estimation based on the time of arrival of the initial blink message at each AN. A huge benefit of TDOA is the fact, that the TAG only needs to send one blink message per timeinterval and will not have to communicate with every AN separately, as in TWR. This can be seen in figure 3.2.

For TWR the overhead grows enourmous with every anchor and every TAG that

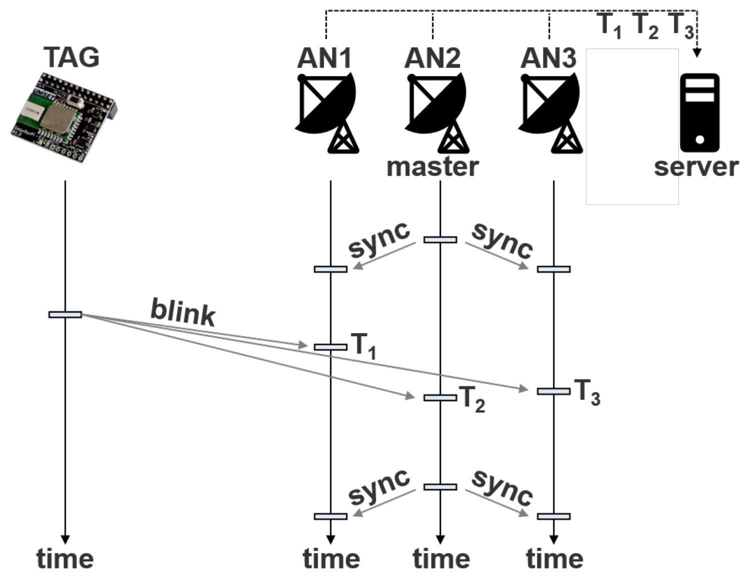

FIGURE 3.2: Typical Setup for Time Difference of Arrival communication

is added. For TDOA hundrets of TAGs can be tracked, with only proportional overhead growth and much lower energy consumption for the TAG.
Number of messages sent in one iteration:
TWR: $3 * n_t * n_{an}$
TDOA: n_t
Where n_t is the number of TAGs and n_{an} is the number of anchor nodes [21].

3.3.3 Multilateration

Multilateration is a mathematical method to calculate a position using three or more known distances. It is an extention of trilateration, where only three distances are used. Triagulation and trilateration use the mathematical concepts of triangles to find unknown lengths. Triangulation was already mentioned by the greek mathematician Thales, who used this concept for finding out the height of ancient egypt pyramids [13]. It was also used for cartography purposes, where angles between fixed points were measured and heights and distances could be calculated. Although trilateration and triangulation use the same mathematical triangle concept, they have one defined difference: We call it triangulation, when angles to anchor positions are measured, otherwise - when distances to anchors are measured - it's called trilateration. As it was easier to measure angles than distances in the past, triangulation was more often used. With modern electronic devices, it is more common to determine distances, rather than angles.

Figure 3.3 shows how trilateration is used for positioning. With the known distance to every AN, a circle with radius of this distance can be drawn around every AN. These circles do only have one common intersection point, that is where the TAG lies. However, this is a theoretical and idealized scenario, where every range can be determined accurately. In real applications, the ranges are not exactly calculated, what leads to the fact that we will not only get a single point for the calculated position, but several points, especially when we use more than three ANs.

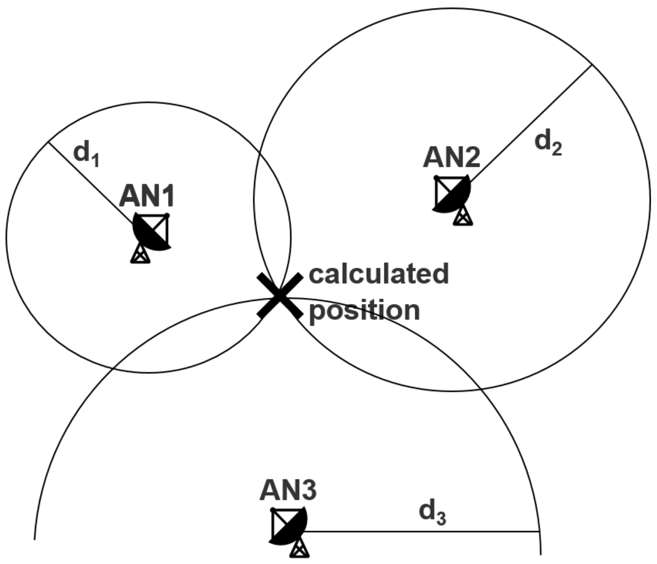

FIGURE 3.3: Graphical illustration of the trilateration concept.

3.4 Movement Detection

Indoor positioning is often reduced to two dimensions, such that the movement vector is also a two dimensional construct. A movement vector Mv_t can be stored as (X_t, Y_t) cartesian coordinates or using a tuple of heading direction angle and stride length (θ_t, ℓ_t). Both of θ and ℓ can be calculated by the IMU readings, where several noises occur such that the heading direction is statistically described as:

$$\theta_t = \hat{\theta}_t + \theta_{bs,t} + \theta_{be,t} + \epsilon_{\theta,t}, \tag{3.6}$$

with $\hat{\theta}_t$ as the actual heading orientation, $\theta_{bs,t}$ as a sensor bias introduced by uncalibrated sensor readings, $\theta_{be,t}$ as an environmental angular bias due to magnetic field disturbances and $\epsilon_{\theta,t}$ as a measured random error. The heading direction can be calculated by using the geomagnetic field of the earth. The formula $\theta = atan(\frac{mag_x}{mag_y})$ can be used to obtain the heading direction, where mag_x and mag_y are the magnetic field sensor readings in X direction, respectively in Y direction.

Whereas the heading direction is obtained by measuring absolute values of magnetic field energy, the accelerometer data is a relative quantity, that is used for stride length determination. Dealing with relative values, measured errors propagate over time, so it is almost impossible to use relative quantities (e.g. acceleration) to calculate absolute quantities (e.g. distance) over a longer period of time. To address this, an absolute quantity is needed to correct long term errors. Velocities from previous time slots can be used for this. As we can not assume the acceleration to be constant during a whole time slot, the movement length approximation can be defined as:

$$\ell_t = \hat{\ell}_{t-1} + \sum_{i=0}^{N}([(\hat{a}_i + a_{bs,i} + \epsilon_{a,i}) * \Delta t_i] * (N - i)) * \Delta t + \epsilon_{\ell,t}, \tag{3.7}$$

where $\hat{\ell}_{t-1}$ is the actual movement length of time period $t-1$, $\hat{a}_i$ is the actual middle acceleration during the i-th of N time slots in time period Δt, $a_{bs,i}$ is an other sensor bias due to uncalibrated sensor readings and $\epsilon_{a,i}$ as well as $\epsilon_{\ell,t}$ are measured random errors in acceleration and distance respectively.

3.5 Data Fusion for Localization

When several different sources of data are collected and brought together in one algorithm, we call it data fusion. There are several common ways of fusing data, the most widely used are filtering approaches such as the Kalman Filter [14] and the Particle Filter. The focus of this work lies on a particle filter approach, also known as Monte Calro Localization (MCL), which is often used fo indoor positioning. It combines various noisy measurements to estimate the system state and minimize errors. To introduce the particle filter, we explain its three main phases and the corresponding inputs.

3.5.1 Particle Filter: Prediction Phase

In the prediction phase, every particle is repositioned. At time t, each particle has a state vector that is defined as follows:

$$X_t = [x_t, y_t, x_{t-1}, y_{t-1}], \tag{3.8}$$

where (x_t, y_t) corresponds to the Cartesian coordinates of the particle at time t and (x_{t-1}, y_{t-1}) at time $t-1$ respectively. The repositioning of the particles is often done randomly, however, it could also be done according to the movement vector or other sensor-based values. If present, floorplan restrictions are applied in this phase, whereas movements through walls are not permitted, they lead to another prediction iteration for that particle.

3.5.2 Particle Filter: Observation Phase

In the observation phase the associated particle weight w_t^i is recalculated for every particle, since the weight does not anymore correspond to the current position. For each anchor node, we have an obtained distance measurement d_t^j, which itself consists various errors. Statistically it can be described as:

$$d_t^j = \hat{d}_t^j + d_{be,t}^j + \epsilon_{d^j,t}, \tag{3.9}$$

where $\hat{d}_t^j$ is the actual distance to node j, $d_{be,t}^j$ is an environmental bias due to local conditions and $\epsilon_{d^j,t}$, is a measured random error.

The weight is updated corresponding to the likelihood of the range observations conditioned on each particle $p(Zd_t|X_t^i)$ at time t, respectively the likelihood of the motion observation conditioned on each particle $p(Mv_t|X_t^i)$ at time t. With the defined observation vector $Zd_t = [d_t^j], j = 1...N$, at time t, where N is the number of ANs.

Then, the probabilities are determined as:

$$p(Zd_t|X_t^i) = p(d_t^j|X_t^i) \tag{3.10}$$

and

$$p(Mv_t|X_t^i) = p(M_{x,t}|X_t^i) * p(M_{y,t}|X_t^i). \tag{3.11}$$

In addition, the zone probability is:

$$p(y_t|X_t^i) = p_{tot}(y_t|z_t^i), \tag{3.12}$$

where y_t is the observed fingerprint at time t and z_t^i the current zone of particle X^i. In order to avoid confusion between different likelihoods used in this work, hereafter we refer to $p(d_t|X_t^i)$ as the ranging likelihood, $p(M_t|X_t^i)$ for the motion likelihood, $p(y_t|X_t^i)$ for the zone likelihood and $p(Z_t|X_t^i)$ as the overall likelihood.

The associated weight w_t^i of each particle is given by ranging as well as by motion information. A particle at the current position (x_t, y_t) with low probability to observe d_t^j in its position will be assigned a small weight. Additionally a particle that moved in x-direction by $x_t^i - x_{t-1}^i$ with low probability to observe the movement $M_{x,t}$ will also be assigned a small weight. Analogue for the movement in y-direction. That leads to the fact that particles with large weights will have a stronger effect to the determination of the state of the system. We assume that all these likelihoods - the ranges to each AN as well as the movement in direction x, y - are statistically independent from each other. Therefore, the overall likelihood is defined as:

$$p(Z_t|X_t^i) = \prod_{j=1}^{N} p(\hat{d}_{j,t}|X_t^i) * p(\hat{M}_{x,t}|X_t^i) * p(\hat{M}_{y,t}|X_t^i) * p(y_t|X_t^i), \tag{3.13}$$

where $\hat{d}_{j,t}$ is the measured distance to the AN j at time t and $\hat{M}_{x,t}$ is the measured motion in x-direction in timeinterval t, respectively $\hat{M}_{y,t}$ in y-direction and y_t is the measured RSS fingerprint.

The individual likelihood for the range observation can then be expressed as:

$$p(\hat{d}_{j,t}|X_t^i) = \frac{1}{\sqrt{2\pi\sigma_j^2}} * exp(\frac{-[\sqrt{(x_t^i - x_j)^2 + (y_t^i - y_j)^2} - \hat{d}_{j,t}]^2}{2\sigma_j^2}), \tag{3.14}$$

where (x_j, y_j) are the known coordinates of the j-th AN. Whereas the individual likelihood of the motion observation in x-direction (analogue for y-direction) is expressed as follows:

$$p(\hat{M}_{x,t}|X_t^i) = \frac{1}{2\pi\sigma_{Mx}^2} * exp(\frac{-[(x_t^i - x_{t-1}^i) - \hat{M}_{x,t}]^2}{2\sigma_{Mx}^2}) \tag{3.15}$$

3.5.3 Particle Filter: Resampling Phase

The resampling phase is an essential component of a particle filter implementation, although it is a computationally expensive step. In the resampling, particles with low assigned weights are repositioned at identical positions as particles with high associated weights. This means, that after the repositioning of the prediction phase and after the weight calculation in the observation phase, a resampling in systematic manner is done. This resampling relies on the overall likelihood $p(Z_t|X_t^i)$, which means that every kind of likelihood is taken into account for this step. After repositioning the particles with low weights (and updating their weight), all weights are normalized to obtain in the next step the weighted center of all particles, which corresponds to the estimated position.

3.6 Ultra Wideband Radio Technology

Ultra wide-band (UWB) is a radio technology in use for military and industrial communication, positioning and collecting sensor data. Unlike other communication

TABLE 3.1: Relationship of Pulse Rates and Communication Distance

Pulse rate[pulse/s]	**Bit rate**[Mbit/s]	**Range**[m]
1’000’000	~40	~100
500’000’000	~100	~10
1’000’000’000	~500	4-10

technologies, UWB occupies a wide area of frequencies instead of just covering a small frequency spectrum. As shown in figure 3.4, UWB spans over a spectrum of more than 500 megahertz (MHz) that lies within the range of 3.1 gigahertz (GHz) and 13.6 GHz. UWB opperates with less energy compared to other communica-

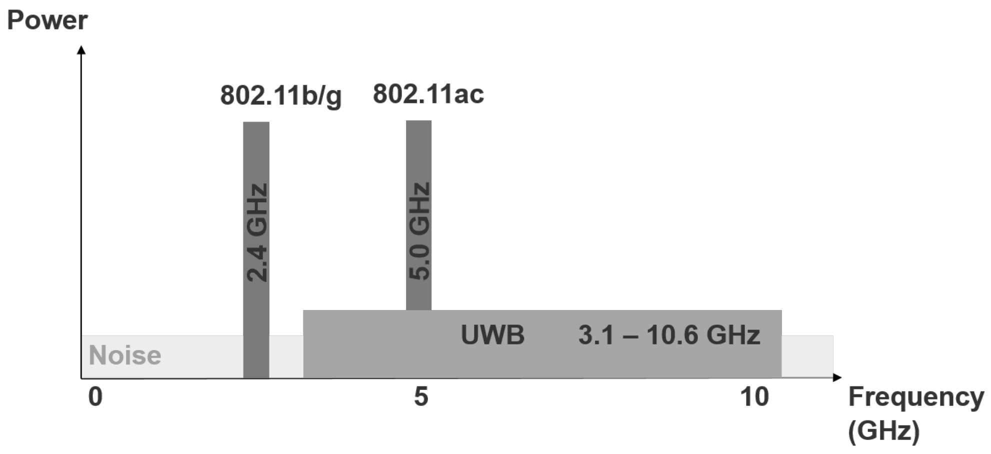

FIGURE 3.4: Comparison of Wi-Fi (802.11) and UWB frequencies.

tion like Wi-Fi. However, the main difference between UWB and conventional radio transmissions is the underlying modulation technique. UWB transmits data by generating short radio energy at specific times instead of varying frequency and phase of sinus waves. In addition to the pulse position, the pulses can carry information either by their polarity, their amplitude or by using orthogonal pulses. A single pulse is kept as short as possible, such that more than 100 Million, sometimes even continuous streams with more than 1 Billion pulses per second are generated. As single pulses can be registered and identified by the receiver, UWB devices are able to determine very exact ToFs such that distance estimations can be done to high resolution.

The pulse rate highly influences the transmission rate of an UWB communication. The pulse frequency varies between 1 million pulses per second to over 1 billion pulses. Devices often support different operation modes, such that the number of pulses can be configured. However, with a higher pulse rate, the transmitting distances decreases, such that a trade-off between datarate and communication distance occurs. In table 3.1 the approximate correlation between pulse rate, bit rate and range is shown [10]. In a cluttered environment, especially for non line of sight communications, the possible communication distance decreases very fast. This has to be concerned for indoor applications, as line of sight is very rare in an indoor scenario.

Chapter 4

Localization System Overview

One of the key contributions of this work is the server-bases localization architecture. In this chapter we present our concrete system design. We provide an overview of the different physical components in the first part, following a second part where we introduce the components of the localization algorithm.

4.1 Localization System Architecture

Our proposed localization system consists of various components, such as a localization server, a target device, several UWB anchor nodes and WiFi access points. An overview of all these components can be seen in figure 4.1.

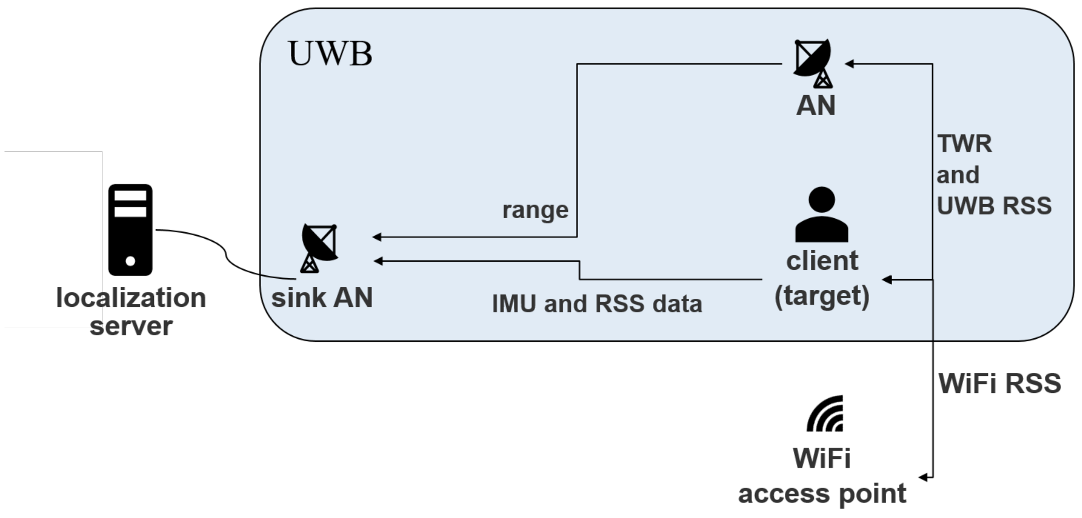

FIGURE 4.1: Overview of the components in our localization system.

On the localization server most of the computational workload is handled. The localization algorithm runs on the server, it requests periodically input data from the other system elements and processes the data as soon as it arrives. The server updates the system state (e.g. particle positions, velocity..) and performs the position estimation in every timestep. Additionally a graphical illustration of the real time position on the floormap is available on server side.

The client is the target device that is localized. It is equipped with different IMU sensors, at least with a 3D accelerometer and a 3D magnetometer. With the sensor readings the client performs a computation of the heading direction and the change in velocity. With the equipped WiFi module and the UWB transmitter, it collects

RSS data of all WiFi access points and UWB anchor nodes in range. The heading direction, velocity change and RSS fingerprint is transmitted from the client to the localization server.

Ultra wideband anchor nodes are distributed over the area of interest to cover it homogenously. One anchor node is specified as a sink AN, it is connected to the localization server with a wired ethernet connection. The sink AN ensures the communication via UWB to the other ANs and the client device. It receives IMU and RSS data from the client and forwards it to the server. When triggered from the sink AN, the other anchor nodes perform two-way ranging to the TAG and forward the result via sink AN to the server.

WiFi access points are as well spread over the area of interest. They fulfill only one purpose, they emit Wi-Fi radio signals that are detected by the client in order to measure the received signal strength.

4.2 Localization Algorithm

The particle filter localization algorithm consists of several tasks relying on different data inputs. An overview of the particle filter and its fused information is illustrated on figure 4.2. The four used data types are IMU sensor data, floorplan information, range estimations and RSS fingerprints.

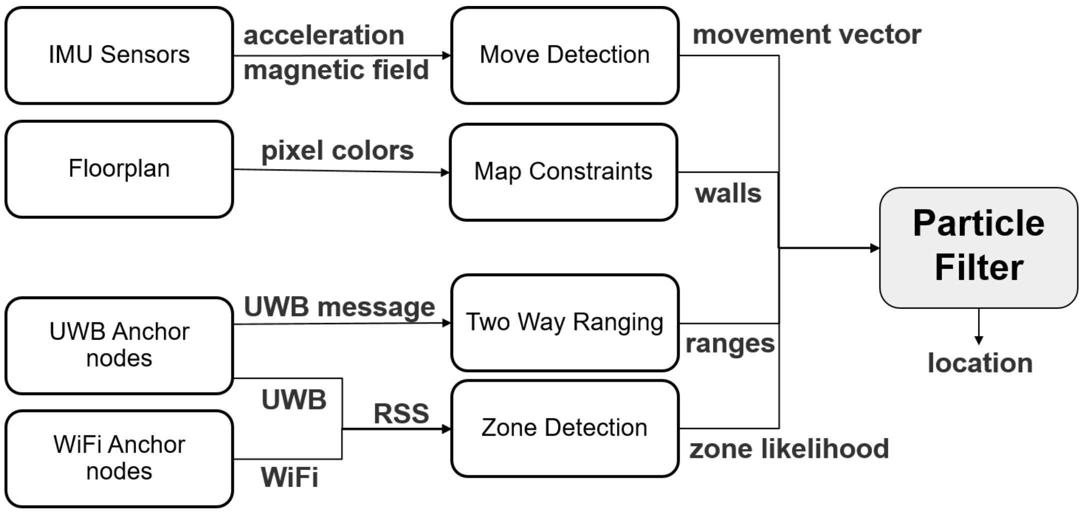

FIGURE 4.2: Overview of the localization algorithm components.

The IMUs are measuring acceleration and magnetic field energy to obtain stride length and heading direction. In the movement detection the stride length and the heading direction are converted to a movement vector, containing X and Y Cartesian coordinates in the buildings coordinate system.

Floorplan information is preprocessed from a given floorplan image. Based on the given information, a map of allowed and not allowed positions is generated, in order to quickly check whether a given particle position is allowed or not.

Ranging information is obtained by two way ranging to every anchor node. The anchor node positions are preliminary stored in the particle filter, such that it is sufficient to process a tupel (AN, range) of anchor node and range measurement.

The zone detection is fed with UWB and WiFi received signal strengh data. Observing the concrete fingerprint of RSS data, for each zone a dedicated probability for being in that zone is computed.

In the particle filter the different inputs are fused. The prediction phase takes only the floorplan constraints into account, whereas for the observation phase the other three inputs - range estimation, zone likelihood and movement vector - are used to determine the weight of a certain particle.

Given the position and the weight of each particle, the system calculates the weighted sum over the particle positions to obtain the localization estimation. The construction of the weight for each particle ensures that more likely positions are stronger taken into account to determine the location than particles with positions that are more unlikely.

Chapter 5

Localization System Implementation

In this section we explain the hardware and software components of our system architecture in detail, then we introduce the implementation of the different localization algorithm components and we describe the underlying UWB communication.

5.1 Localization Components Hardware

The four different hardware components - server, client, ANs and APs - that were used in our implementation can be seen in figure 4.1.

Our proposed algorithm was running on a commodity laptop, which was used as our server, connected via ethernet to a sink anchor node. The laptop was a low to mid-end commercial notebook with fairly limited computational power.

The target device was a Raspberry Pi Model B [17] with a 1.2 GHz 64-bit CPU running the operating system Raspian. On the 40 pin GPIO, it was equipped with a Sequitur InGPS Lite Tag Chip from UNISET Company to enable UWB communication as well as acceleration and magnetic field measurements.

The sink node and the other anchor nodes were also Raspberry Pi Model B with the same specifications. However, they were not equipped with a TAG Chip, but with a Sequitur InGPS Lite Anchor Chip, exclusively enabling UWB communication.

For the WiFi access points we used several commercial commodity access points from D-Link (D-635 and DAP-2553).

The detailed specification of each component is summarized in table 5.1.

5.2 Localization Components Software

The localization algorithm was configured with a minimal update frequency of 0.7 seconds. This means, that the system triggered all input sources and performed all three particle filter phases, such that every 0.7 seconds a position estimation was provided. However, in some cases the update time was longer, when it took more time to fetch data from the data sources. The position estimation was done based on 100 particles.

The motion vector was calculated by using the velocity of the device in the last system update. This velocity was turned to the measured heading direction and adapted by the obtained acceleration. As the sample rate of the accelerometer sensor was 10 Hz, so for each system state update we had seven or more descrete acceleration measurements, we stored these measurements in the client device until the server requested the movement vector. As the acceleration sensor - depending on

TABLE 5.1: Hardware Components

Device	Model
Server	**Model:** HP EliteBook **CPU:** 2.30 GHz Intel Core i5-5300U **Architecture:** 64-bit **OS:** Windows 10 Enterprise **RAM:** 8 GB
Target	**Model:** Raspberry Pi Model B **CPU:** Quad Core 1.2GHz **Architecture:** 64-bit **OS:** Raspbian 4.14 **RAM:** 1 GB **WLAN:** WiFi b/g/n **Extension:** Sequitur Pi (InGPS Lite Tag)
Anchors	**Model:** Raspberry Pi Model B **CPU:** Quad Core 1.2GHz **Architecture:** 64-bit **OS:** Raspbian 4.14 **RAM:** 1 GB **WLAN:** WiFi b/g/n (not used) **Extension:** Sequitur Pi (InGPS Lite Anchor)
Ultra WideBand Extention	**Model:** Sequitur Pi (InGPS Lite) **Ultra WideBand:** IEEE 802.15.4a **IMU (tag only):** 3D-Accelerometer, 3D-Magnetometer **Firmware:** Sequitur InGPS Lite Anchor/Tag (from UNISET)

pitch and roll of the device - had a huge non-zero mean noise, we decided to not use the accelerometer data directly, but to use the change in acceleration.To gather the change in acceleration we fed the sensor data into two low pass filters with different parameters, one with a high adaption of 0.98 and one with a low adaption of 0.03, and only took their difference into account. This means, we calculated the difference between a_t^1 and a_t^2, which were calculated as in 5.1 and 5.2:

$$a_t^1 = \hat{a}_t * 0.03 + a_{t-1}^1 * 0.97 \tag{5.1}$$

and

$$a_t^2 = \hat{a}_t * 0.98 + a_{t-1}^2 * 0.02 \tag{5.2}$$

with $\hat{a}_t$ as the measured acceleration in timeslot t and a_{t-1}^1 respectively a_{t-1}^2 the low pass filtered acceleration results in timeslot $t-1$ of the first, respectively the second low pass filter.

We read the floorplan information from an image of the floorplan. The image data was then stored in a matrix with values 0, if the position was not allowed and any other value bigger than 0, when the position was an allowed position. For checking if a position was reachable, we just checked all values in the matrix that were passed when taking the direct path between the old and the new position.

For gathering ranging information, the Sequitur InGPS Lite firmware provided a two way ranging method. The range estimation of two nodes was triggered by the application programming interface (API) command $CLIENT\_GET\_RANGE$ (50). In our application we sent the command to the anchor in order to minimize the communication of the TAG. The flow of actions related to this API is a even more simplified version of the message exchange indicated in figure 3.1. In our case, the request message performed by the server started the TWR conversation via UWB between AN and TAG. The AN sent only one ranging request to the TAG, which immediately responded. By observing the difference between the time instants related to the transmission of the request packet and the reception of the response packet, the AN directly determined the RTT and thus the range. Finally an answer message with the range was reported from the AN to the server and no messages were reported from the TAG to the server.

The zone indication fused Wi-Fi and UWB RSS in an enhanced ensemble learning model. In our zone indication a set of independent individual machine learning algorithms were fed with the same fingerprint data. Every of the machine learning algorithms performed a zone prediction and assigned a likelihood to every zone. The likelihood respresented the probability of observing the given RSS fingerprint while being in this zone. We assumed that for every of the machine learning algorithms these probabilities were statistically independent, thus the probabilities returned by our ensemble learning algorithm were just the multiplied results of the single machine learning algorithms.
We used these three completely different ML algorithms: Decision Tree Learning with gini impurity [25], K nearest neighbor (KNN) classification [11] and a soft voting classifier using gini, KNN and gaussian naive bayes [18, 9] algorithms. We selected these three algorithms out of half a dozen machine learning algorithms, because those three had the best classification results in the test data. For the testing data the three machine learning algorithms had a correct prediction rate of 80 to 99 percent.

TABLE 5.2: Spectral Occupation of Predefined Channels.

Channel Number	**Central Frequency**[MHz]	**Bandwidth**[MHz]
1	3494.4	500
2	3993.6	500
3	4492.8	500
4	3993.6	1300
5	6489.6	500
7	6489.6	1100

5.3 Ultra Wideband Communication

The radio module of Sequitur Pi board was not only used to evaluate the ToF, but also to transmit data to the server via sink anchor node, in order to obviate the need for additional communication hardware. As UNISET is a commercial company, they do not provide full information of the underlying transmission techniques. Nonetheless in the following, we mention the known parts and the configuration parameter.

Sequitur InGPS Lite enables single-hop wireless communication with the UWB interface between neighboring nodes of the same network. The radio module supports different user-selectable frequency bands between 3.5 GHz and 6.5 GHz. There are six different operation modes to change the spectral occupation, listed in table 5.2.

The data rate can be changed to three preset values of 110 kilobit per second (kbps), 850 kbps and 6.8 megabit per second (Mbps). All nodes have to operate in the same radiomode and frequency band to communicate correctly. In general, a lower data rate allows lager operating distances between the nodes. The transmission power of the radio module could be selected between 1 and 63, whereas 63 is the highest value. Every number increases the transmitting power of 0.5 dB. The default pulse repetition frequency (PRF) is assumed to 64 MHz for all the channels. The underlying modulation techniques are not indicated in the specifications [20, 19].

In our implementation the UWB transmitter of all UWB devices operated in radio mode 2 with a datarate of 850 kbps, they were configured to use channel 4 with a central frequency of 3993.6 MHz and an occupied sprectrum of 1300 MHz. The transmission power was set to a maximum uf 63 and the PRF to the default value of 64 MHz. Before starting the experiments, a 15 meter calibration with 1000 measurements per device was made as described in the beginners guide [19].

Chapter 6

Performance Evaluation

In this section we explain the setup of our two experiment scenarios and we present the positioning results of our experiments in detail.

6.1 Experiment Setup

We tested our implementation in two complex indoor scenarios with trajectories through numerous of rooms on one floor in a real building of the University of Bern. The first scenario used an area of $715m^2$ and the second scenario $358m^2$. We distributed the UWB anchor nodes over several rooms to cover the area of interest homogenously. The exact position is indicated in the floor plan of figure 6.1 for the first

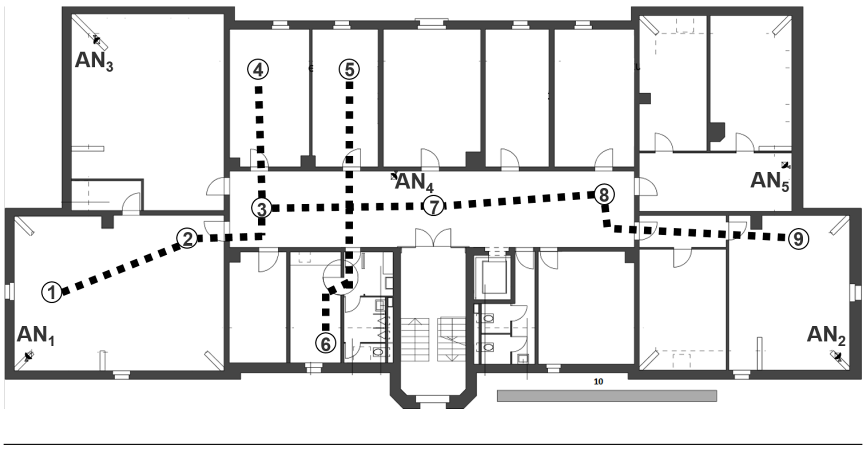

FIGURE 6.1: Trajectory 1 and distributed ANs in scenario 1 on the floor map (with distance reference of 10m).

scenario and indicated in figure 6.2 for the second scenario. In both scenarios the target was hold in the hand of a pedestrian at the starting point of the trajectories, when the experiments started. The pedestrian walked along the given trajectory path, as soon as he passed a predefined checkpoint the current position estimation was registered.

We defined four different trajectories for the first scenario. Each with five to nine checkpoints. Trajectory 1 is indicated in figure 6.1, the other three trajectories can be seen in appendix A. For the second scenario we used a fifth trajectory that covered almost the whole area, it is shown in figure 6.2.

We repeated the experiments five times, so we analized 145 checking points in scenario 1 and 40 checking points in scenario 2. The localization error was determined by the euclidian distance between the systems position estimation and the real position of the checking point.

For the zone indication we defined 14 zones, which corresponded to 13 different

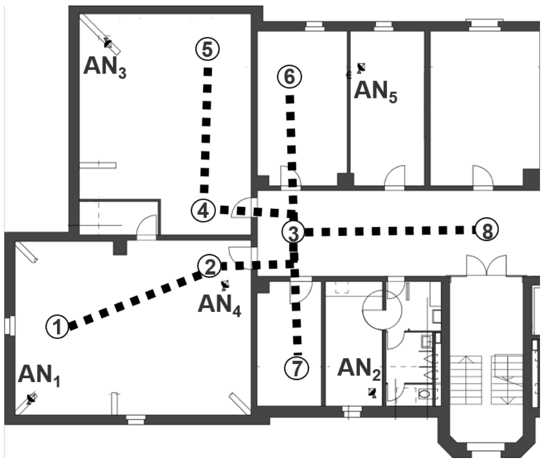

FIGURE 6.2: Trajectory 5 with improved anchor positions.

rooms, where only one room (i.e. the corridor) was split in two zones. The room definitions can be seen in figure 6.3. For scenario 1 we collected data in all zones, whereas for the smaller scenario we only collected data in the rooms 1, 2, 3, 5, 6 and 7, as only these were in the area of interest. The received signal strength of all five UWB ANs as well as of eight Wi-Fi access points was taken into account. The machine learning algorithms were fed with RSS difference to a given pivot RSS. The fourth AN and the neighboring WiFi access point were acting as pivot. In the offline phase of the fingerprinting, we collected around 700 fingerprint measures per room. Half of them were collected randomly passing the room and half of them were collected while systematically walking through the whole area of the room.

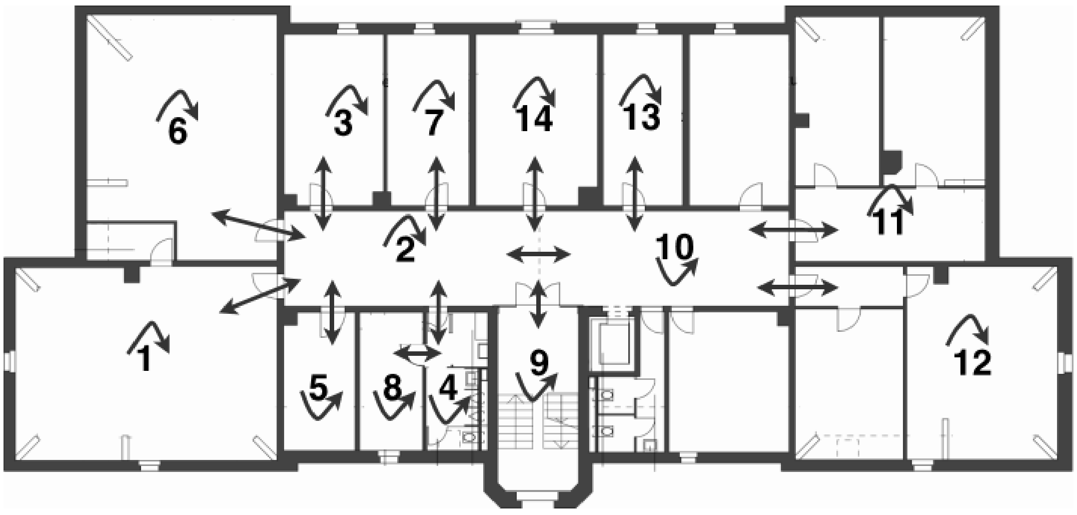

FIGURE 6.3: Zone definition and transitions between zones

6.2 Experiment Results

Experiments were first conducted in scenario 1, then we improved the anchor node position and conducted the experiments in scenario 2. As a reference we tested the same two scenarios with a different localization algorithm commercially proposed by UNISET Company, called Sequitur InGPS Lite. In the following we compare our particle filter results to the Sequitur system results.

6.2.1 Positioning Results for Scenario 1

In the following the results of our algorithms with a wide distance between anchor nodes (scenario 1) are shown. The results are shown as arithmetic means of the five probes we registered. In the first trajectory, the arithmetic mean (hereafter often called average) distance error over all checkpoints was 1.62 meter for the particle filter and 1.75 meter for Sequiturs commercial system. Looking at figure 6.4, we see that the errors are often smaller than 1.5*m*, however, there are some checkpoints with very low accuracy. This is also emphasized by comparing the arithmetic mean error to the median error of 1.12*m* for PF and 1.13*m* for Sequitur, which are significantly lower than the arithmetic means. The measurements for trajectory two looked rather similar but with a higher error. The arithmetic mean error for the PF was 2.39*m* and for Sequitur 2.35*m*. The median errors were again a lot more accurate with 1.59*m* for PF and 1.16*m* for Sequitur. The results for trajectory 3 and 4 were rather sim-

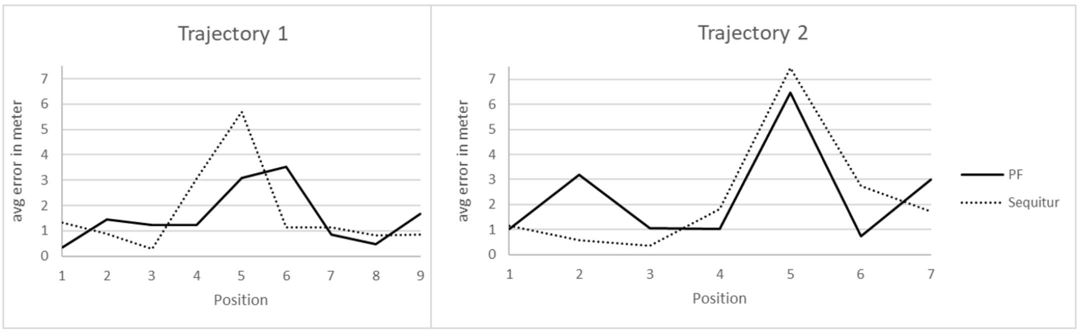

FIGURE 6.4: Graphs of measured distance errors at each checkpoint in trajectory 1, respectively trajectory 2.

ilar, however, the peaks observed in figure 6.5 were not as extreme as for the first two trajectories. As above, the average errors in the third trajectory of 1.23*m* (PF) and 1.94*m* (Sequitur) were also mentionable higher than the medians of 0.95*m* and 0.80*m*. In the last of these four trajectories no big outliers were stated. Nonetheless the average errors of 1.79*m* and 1.55*m*, as well as the median errors 1.24*m* and 1.26*m* for the PF and Sequitur, were still not as accurate as intended.

Having a closer look at table 6.1, we see that there are very big differences between different trajectories. The particle filter and the Sequitur system have similar accuracies, for some trajectories one of the algorithms is better, for other trajectories the other performs better. Possible reasons for these volatile results are discussed in the upcoming subsection 6.2.3 about result comments.

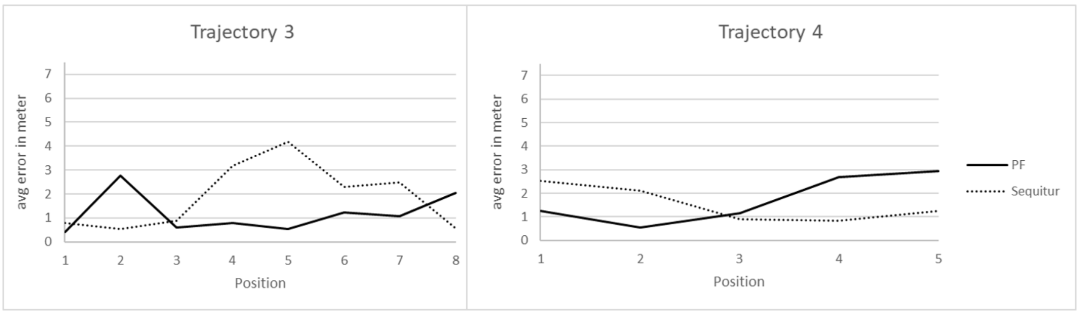

FIGURE 6.5: Graphs of measured distance errors at each checkpoint in trajectory 3, respectively trajectory 4.

TABLE 6.1: The Arithmetic Mean of Errors in Trajectory 1 to 4 (in meter).

Trajectory	PF	Sequitur
T1	1.72	1.75
T2	2.39	2.35
T3	1.23	1.94
T4	1.79	1.55
Total 1-4	**1.73**	**1.93**

6.2.2 Positioning Results for Scenario 2

For trajectory 5 the distances between the anchor nodes were shortened. The exact setup was indicated above in figure 6.2. Equally as before the results are shown as arithmetic means of the five test repetitions.

The results in this setup were way better than with wider ANs. With an average error of 0.49m for the PF and 0.48m for Sequitur, this setup outperformed the other trajectories with both algorithms. Even the median errors - with 0.44m and 0.48m - were not much different, which means that there are no huge differences over the different checkpoints. This can also be seen in figure 6.6, where most of the errors are smaller than 0.80m.

In table 6.2, the mean errors of the low density AN scenario 1 and the high density AN scenario 2 are compared. The results improved significantly with a denser anchor node positioning. The standard deviation (S.D) for scenario 2 was rather small with 0.24m for the PF and 0.27m for the Sequitur system, whereas the 90% accuracy was 0.87m and 0.82m.

TABLE 6.2: The Arithmetic Mean of Errors in Trajectory 5 (in meter).

Trajectory	PF	Sequitur
Total 1-4	1.73	1.93
T5	0.49	0.48
Difference	**-1.24**	**-1.45**

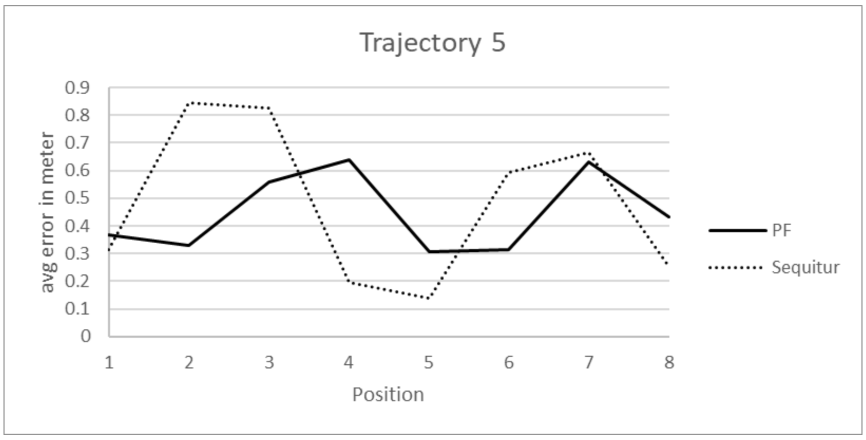

FIGURE 6.6: Graphs of measured distance errors at each checkpoint in trajectory 5.

TABLE 6.3: Table of Results for Trajectory 5.

Algorithm	**Mean error**[m]	**S.D**[m]	**90% Acc.**
PF (Scen 1)	1.729	1.451	3.393
Sequitur (Scen 1)	1.928	2.337	5.544
PF (Scen 2)	0.491	0.239	0.875
Sequitur (Scen 2)	0.484	0.271	0.829

6.2.3 Comments on Test Results

The test results in the experiment setup with wide anchor node distances were not as good as intended. To find the reasons for that, we have to carefully have a look at the underlying implementation and the single checkpoint circumstances. We identified two main causes for bad results, these were:

- Checkpoints outside of AN bounding boxes
- Failing UWB connection due to too wide distances to ANs

A cluttered environment will heavily distort UWB communication and thus affect the RTT used for UWB ranging. As in our testing environment servers with iron racks, desks with computer screens as well as a lot of different equipment was present, this effect should not be neglected.

The AN positions naturally form a bounding box around the environment. The bounding box is the area spanned by straight connections between anchor nodes, as indicated in figure 6.9. Trilateration, also with small distance errors, works well within the bounding box. However even small ranging errors can lead to wrong position estimations outside the bounding box. In our experiment especially the results on checkpoint 5 in trajectory 1 and 2 stand out. The average measured errors of 3.09m for PF and 5.70m for Sequitur during the first trajectory as well as 6.46m and 7.45m during the second trajectory are way higher than for other checkpoints. The fact that both algorithms had troubles estimating this position, leads to the conclusion that the experiment setup was the main reason for the big errors at this checkpoint. Moreover, not only lies this point outside the bounding box, but it was also

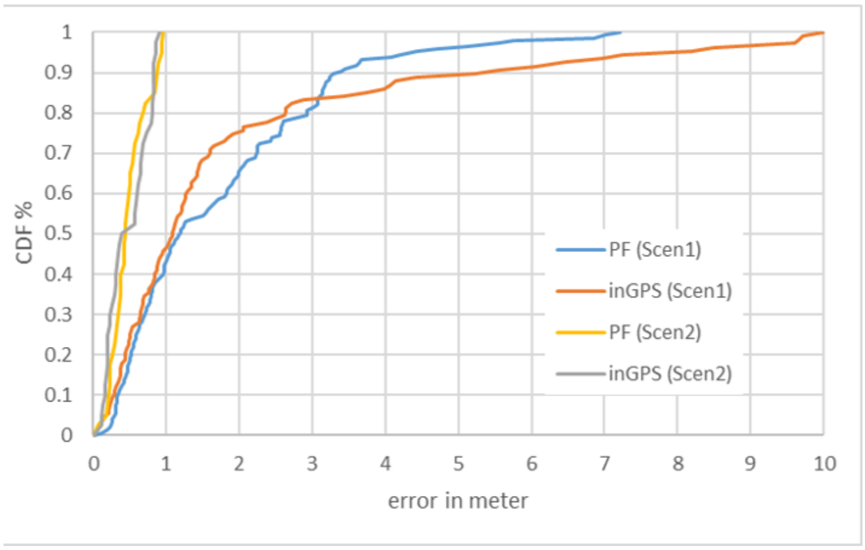

FIGURE 6.7: The cummulative distribution function of the errors.

far away from the nearest ANs without a direct line of sight.

Both algorithms depend on a good established UWB connection, as all the ANs ranging estimations are only transmitted via UWB. Especially in the particle filter many UWB meassages are transmitted, because fingerprinting and IMU data are additionally requested and exchanged. For certain checkpoints the distance to the farest anchor node was even more than 35m in scenario 1, which was too much for a stable UWB connection within our environment. Particularly, the particle filter had troubles receiving enough usable data, as it produced more overhead than the Sequitur system. This led to a high package loss, such that only one or two range measurements per estimation step were taken into account - instead of the possible five - leading to a higher error.

These surprisingly big inaccuracies of both systems prompted the change in our setup, in order to establish better communication between the nodes with less package loss to enforce more data flowing into the particle filter. This was the main reason why we extended our experiments and evaluated the location accuracy in scenario 2.

With the more dense spread of the ANs, the estimations improved a lot. Having a look at our two assumptions above, we can state the following:

- Checkpoints outside of AN bounding boxes had quite good accuracy, adding doubts to this explanation.
- The communication was better, what improved the accuracy a lot and confirmed our assumption.

In the new setup we purposely added checkpoints at the edge or outside the bounding box, as seen in figure 6.10. Especially checkpoint 5 and 8 are not lying within the bounding box, however they still had a quite good accuracy. For checkpoint 5, the average estimation errors were 0.30m and 0.14m, for checkpoint 8 the errors were 0.43m and 0.25m for PF, respectively Sequitur. It made no difference for the tag being outside the bounding box or not. We conclude that with good ranging data it is not very important to stay within the bounding box, however, wrong ranging data leads to bigger errors outside the bounding box.

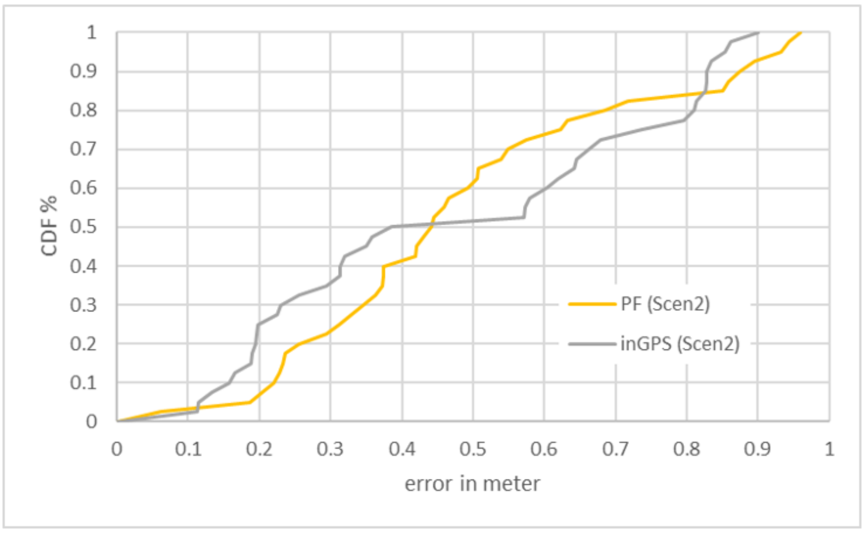

FIGURE 6.8: Zoomed cummulative distribution for scenario 2.

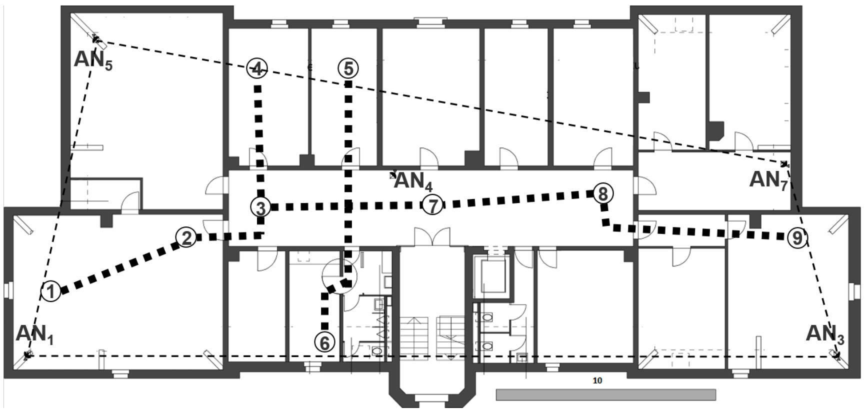

FIGURE 6.9: Indicated bounding box of the anchor node positions for trajectory 1.

A bigger influence on the estimation error had the UWB connection. With a better established connection - in our case with nearer ANs - more UWB messages arrived at the receiver. In each estimation step, more data was fed into the positioning algorithms what caused obviously a more precise performance. For the particle filter the communication is one of the key performance values, because the data is requested serially. This only allowes a limited amount of retransmitting and the communication timeouts have to be kept short. The resulting increased package loss affects the data quality, such that some evaluating steps had to be performed with less data. Obviously the lack of data led to bad results, as the particle filter improves its estimation by fusing many different measurements, what is not possible when a part of the data is not present.

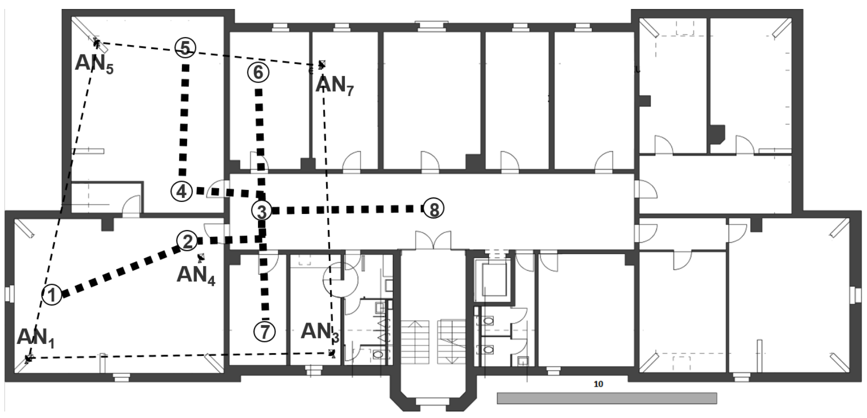

FIGURE 6.10: Indicated bounding box of the anchor node positions for trajectory 5.

Chapter 7

Conclusion and Further Work

In this section we conclude our findings and our work. In the last section we propose which aspects in particular could be addressed in further research.

7.1 Conclusion

In this bachelor thesis an enhanced particle filter fusing UWB radio signals, inertial sensor measurements, physical environmental information and WiFi signals is presented. Running on Raspberry Pi devices, the system achieves high localization accuracy in complex indoor scenarios. With a proper anchor node positioning, the particle filter accomplished an average accuracy of $0.49m$ and a 90% accuracy of $0.87m$. In comparison to smartphone based implementation in previous works of the CDS (avg accuracy of $1.15m$ and 90% accuracy of $1.8m$), our UWB radio signal implementation is promising for the future [5]. The prototype of our implementation could even keep up with the commercial indoor positioning system Sequitur InGPS Lite from UNISET Company. Although we observed a rather good accuracy, we anticipate that the accuracy could be improved even more in further research by optimizing our algorithm according to the points listed in the next section.

7.2 Further Work

In future work, the positioning of the anchor nodes should be evaluated to higher detail. For the implementation itself, we identified the following four main improvements to our particle filter approach to address in further research:

- Seperate UWB communication
- Stability (especially when no data is available)
- Run time performance allowing more particles
- Wall detection deadlock

A key bottleneck in our implementation was the UWB communication, as data was requested directly when it was needed. After each request, this caused a small waiting period, where the system waited for the requested data. In some cases the timeout was reached and the system continued without the requested data. A separate communication unit with anticipated data requests and a proper handling of received messages would definitly improve the data quality and thus the performance.

Due to environmental conditions, a bad UWB connection is very likely to occur. In this case, only a small amount of data was available for the estimation. Especially when none of the ANs responded to the data requests, the system terminated with errors, as obviously it was not able to perform a position localization. For these cases a defined fallback should be taken into account, in order to achieve a stable localization.

The particle filter algorithm, especially with the room recognition, is computationally demanding. All computations are done serially one after another, just like the requesting of data. The estimation run time should be improved by adding contemporal computations and eliminating unnecessary loops. An improvement in runtime performance would also allow to run the algorithm on the same server with more than 100 particles, which could further improve the accuracy.

Finally for some trajectories the wall detection produced a deadlock, when walking around wall edges with moderate speed. In this case the system was not able to reestablish good accuracy without going back to the room, where the deadlock occured. Adding a deadlock recognition that would force a reset of the localization engine could be favorable to eliminate this behaviour.

Addressing these gaps, I am sure that UWB based indoor positioning using particle filter error correction is very promising for the future to outperform other localization technologies.

Appendix A

Trajectories

A.1 Figures of the Trajectories used in the Experiments

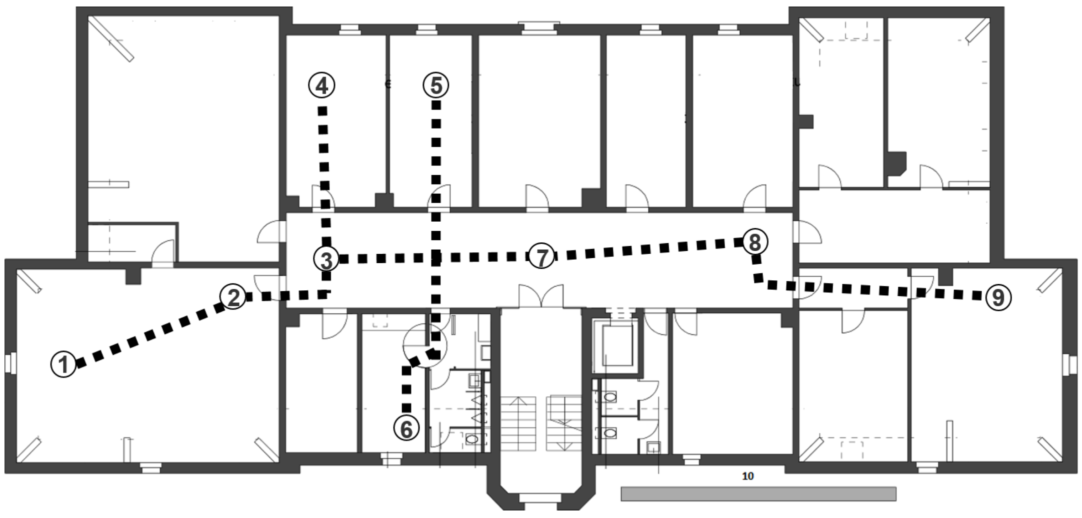

FIGURE A.1: Checkpoints and path of trajectory 1.

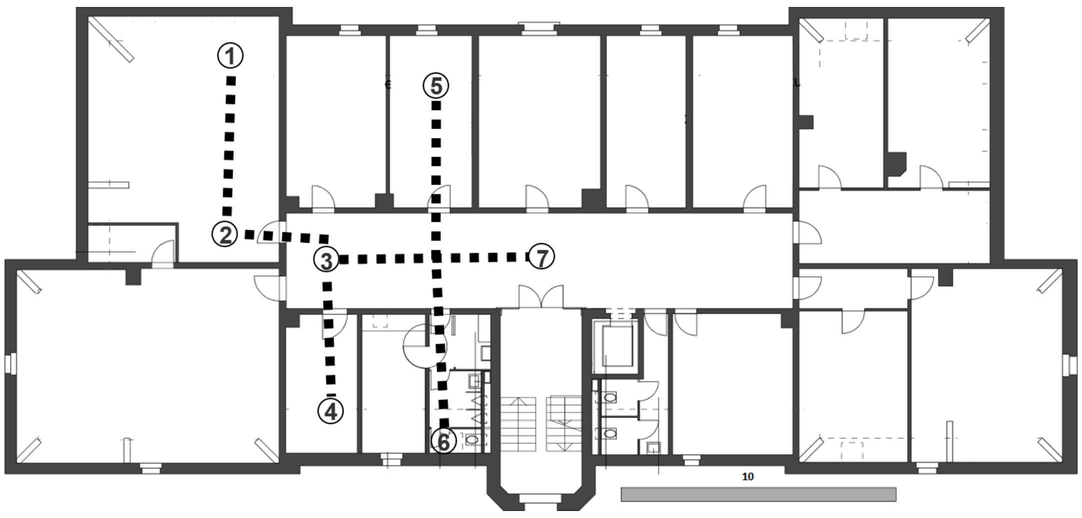

FIGURE A.2: Checkpoints and path of trajectory 2.

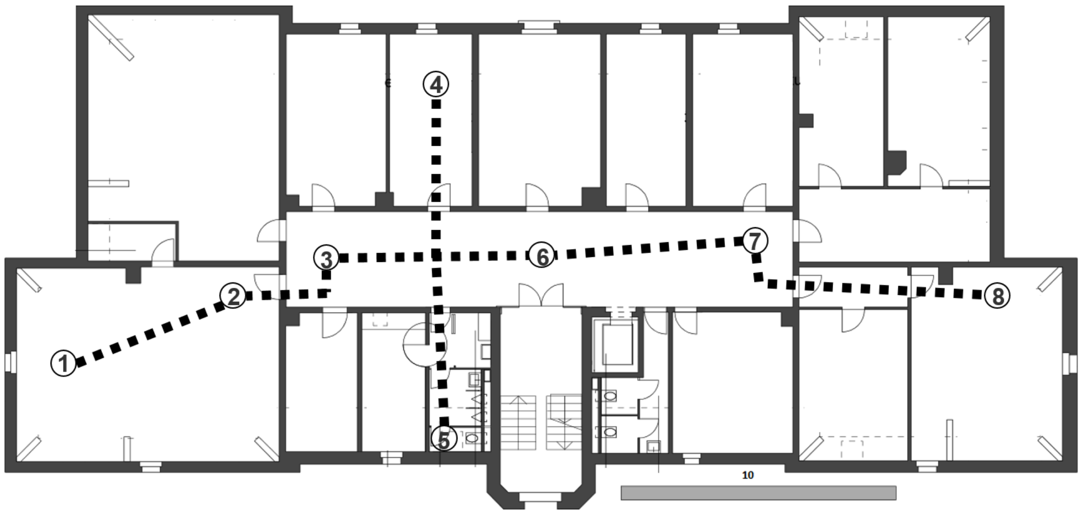

FIGURE A.3: Checkpoints and path of trajectory 3.

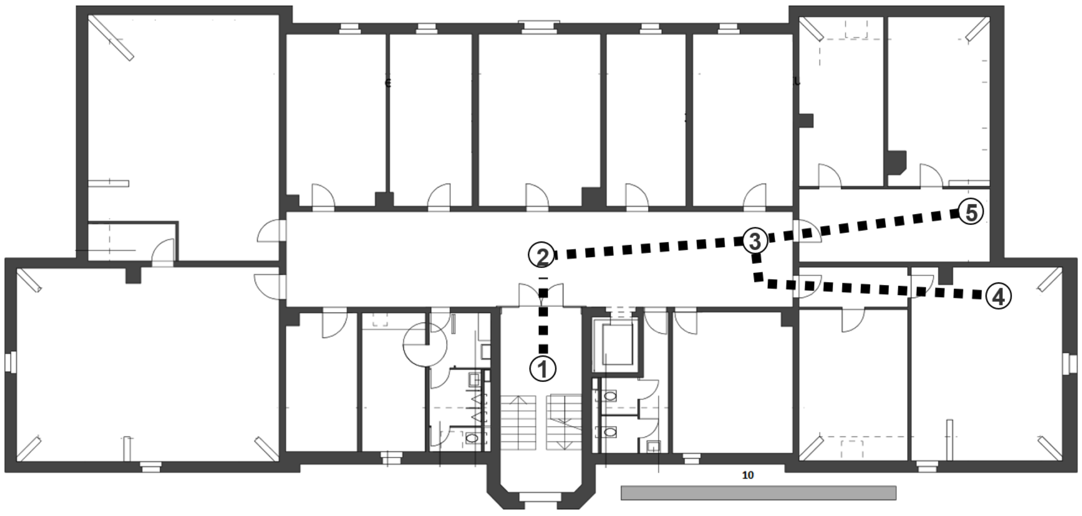

FIGURE A.4: Checkpoints and path of trajectory 4.

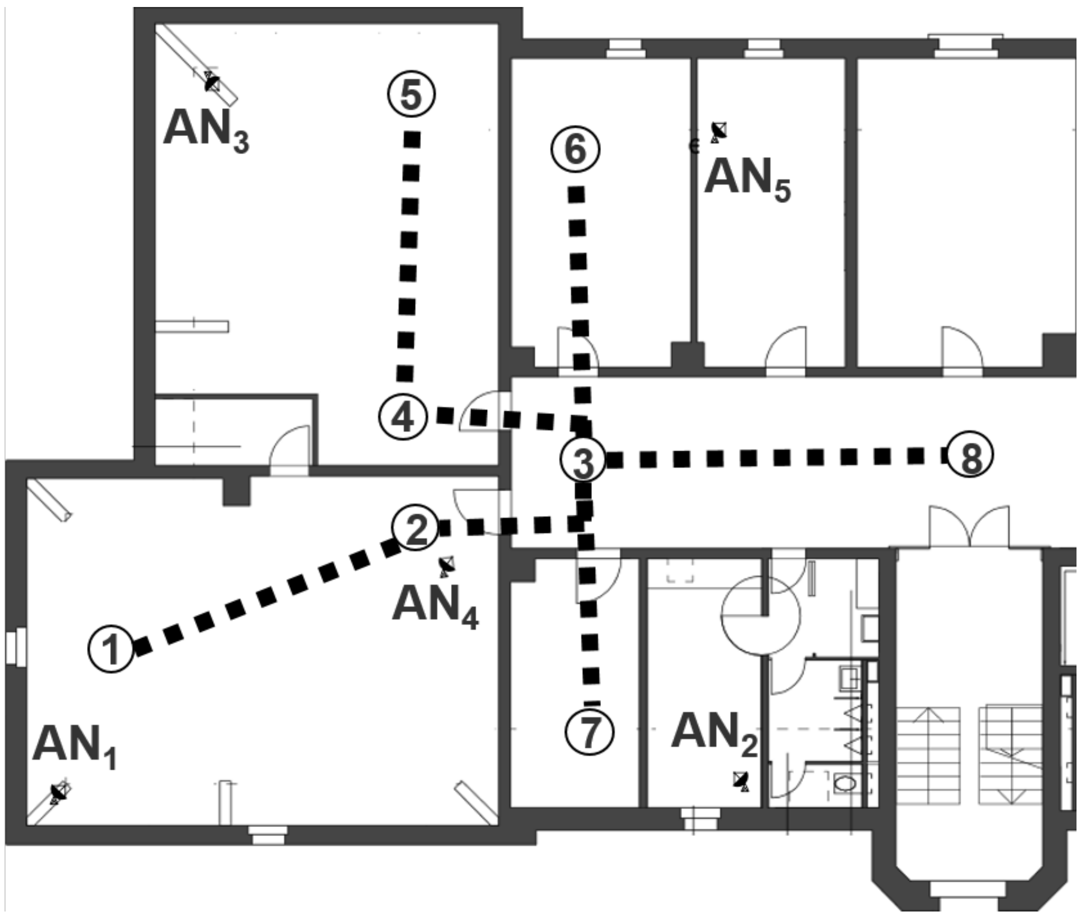

FIGURE A.5: Checkpoints and path of trajectory 5 with anchor positions.

Bibliography

[1] Kaikai Liu et al. "Guoguo: Enabling Fine-grained Indoor Localization via Smartphone". In: *Proceeding of the 11th annual international conference on Mobile systems, applications, and services* (2013), pp. 235–248.

[2] Sarkar Tapan K. et al. "A survey of various propagation models for mobile communication". In: *Antennas and Propagation Magazine, IEEE 45.3* (2003), pp. 51–82.

[3] J. Borestein and L. Ojeda. "Heuristic drift elimination for personnel tracking systems". In: *The Journal of Navigation* (2010), pp. 591–606.

[4] Kerem Ozsoy; Ayhan Bozkurt and Ibrahim Tekin. "Indoor Positioning Based on Global Positioning System Signals". In: *https://onlinelibrary.wiley.com/doi/abs/10.1002/mop.27520* (2013).

[5] J.L. Carrera; Z. Zhao; T. Braun and Z. Li. "A real-time robust indoor tracking system in smartphones". In: *Computer Communications* (2017).

[6] Z. Li; T. Braun and D.C. Dimitrova. "A Passive WiFi Source Localization System based on Fine-grained Power-based Trilateration". In: *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)* (June 2015), pp. 1–9.

[7] Z. Li; T. Braun and D.C. Dimitrova. "A time-based passive source localization system for narrow-band signal". In: *The IEEE International Conference on Communications (ICC)* 39 (June 2015), pp. 4599–4605.

[8] Bruno R. e Delmastro F. "Design and Analysis of a Bluetooth-Based Indoor Localization System". In: *Conti M., Giordano S., Gregori E., Olariu S. (eds) Personal Wireless Communications. Lecture Notes in Computer Science, vol 2775. Springer* (2003).

[9] *Gaussian naive bayes: Naive Bayes documentation*. http://scikit-learn.org/stable/modules/naive_bayes.html. Accessed: 2018-09-05.

[10] *ITU: Characteristics of ultra-wideband technology*. http://www.itu.int/dms_pubrec/itu-r/rec/sm/R-REC-SM.1755-0-200605-I!!PDF-E.pdf. Accessed: 2018-09-29.

[11] *K neighbors classifier: K nearest neighbors classifier documentation*. http://scikit-learn.org/stable/modules/neighbors.html. Accessed: 2018-09-05.

[12] N. Kakiuchi and S. Kamijo. "Pedestrian dead reckoning for mobile phones trhough walking and running mode recognition". In: *Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013)* (2013), pp. 261–267.

[13] Richard Kerner. "The Thales experiment". In: *https://arxiv.org/pdf/1712.06016.pdf* (2017), pp. 1–2.

[14] Adrian Kurt. "Indoor Tracking with Kalman Filters using RSS-based Ranging". In: *Bachelorarbeit Universität Bern* (2015), pp. 4–5.

[15] P. Nagpal and R. Rashidzadeh. “Indoor positioning using magnetic compass and accelerometer of smartphones”. In: *International Conference on Selected Topics in Mobile and Wireless Networking MoWNet* (2013), pp. 140–145.

[16] D. Raniuchi and T. Maekawa. “Robust wi-fi based indoor poitioning with ensemble learning”. In: *IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications* (2014), pp. 592–597.

[17] *Raspberry Pi: Product specification*. `https://www.raspberrypi.org/products/raspberry-pi-3-model-b/`. Accessed: 2018-07-31.

[18] *Scikit-learn: Ensemble methods*. `http://scikit-learn.org/stable/modules/ensemble.html`. Accessed: 2018-09-05.

[19] *SEQUITUR InGPS Lite: Beginners Guide and Application Examples*. SeqInGPSLite BeginnersGuide_R1_2.pdf.

[20] *SEQUITUR InGPS Lite: User Manual and Application Programming Interface*. SeqInGPSLite UserManual_R1_1.pdf.

[21] *Sewio: UWB Technology - Time difference of arrival*. `https://www.sewio.net/technology/time-difference-of-arrival/`. Accessed: 2018-07-23.

[22] *Sewio: UWB Technology - Two way ranging*. `https://www.sewio.net/technology/two-way-ranging/`. Accessed: 2018-07-23.

[23] *Statista: Internet of Things - number of connected devices worldwide 2015-2025*. `https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/`. Accessed: 2018-07-19.

[24] *Vorlesung Computernetze UniBe: Vorlesungsunterlagen Uebertragungsmedien*. Herbstsemester: 2015.

[25] *Wikipedia: Decision tree learning*. `https://en.wikipedia.org/wiki/Decision_tree_learning`. Accessed: 2018-09-05.

[26] *Wikipedia: Symmetrical double-sided two-way ranging*. `https://en.wikipedia.org/wiki/Symmetrical_double-sided_two-way_ranging`. Accessed: 2018-07-24.

[27] M. Youseff and A. Agrawala. “The horus wlan location determination system”. In: *Proceedings of the International Conference on Mobile Systems, Applications, and Services (Mobisys05)* (2005), pp. 205–218.

[28] J.L. Carrera; Z. Zhao and T. Braun. “Room Recognition Using Discriminative Ensemble Learning with Hidden Markov Models for Smartphones”. In: *https://arxiv.org/abs/1804.09005* (2018).