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

September 26, 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 an alternative indoor positioning algorithm based on time of flight measurements of ultra wideband (UWB) signals. In addition, our particle filter approach fuses IMU measurements, floor plan information and WiFi/UWB signal strength fingerprinting, to achieve a high accuracy indoor localization. We evaluated our system, running on Raspberry Pi devices equipped with SEQUITUR Lite UWB transmitters, in complex test scenarios performed in a real indoor environment. 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, it can keep up with the SEQUITUR InGPS Lite system and outperforms previous signal strength implementations, which makes it highly promising for future research.

Contents

Declaration of Authorship **iii**

Abstract **v**

1 Introduction **1**
1.1 Motivation 1
1.2 Contributions 2
1.3 Overview 2

2 Related Work **3**
2.1 Client-based and Server-based Architecture 3
2.2 Fingerprint Localization 3
2.3 Range-based Localization 3
2.4 Pedestrian Dead Reckoning 4
2.5 Combined Localization 4

3 Background Theory **5**
3.1 Client and Server-based Localization 5
3.2 Fingerprinting Localization Technique 5
3.3 Range Based Localization 6
3.3.1 Propagation Models 6
3.3.2 Time of Flight Based Models 7
3.3.3 Multilateration 9
3.4 Movement Detection 9
3.5 Data Fusion for Localization 10
3.5.1 Particle Filter: Prediction Phase 11
3.5.2 Particle Filter: Observation Phase 11
3.6 Ultra Wideband Radio Technology 12

4 System design **15**
4.1 Overview 15
4.2 Setup 16
4.3 Computations on the server 16

5 Implementation and experiment setup **19**
5.1 Hardware 19
5.1.1 Raspberry Pi 19
5.1.2 SEQUITUR Pi board with InGPS Lite 19
5.2 UWB communication and ranging 20
5.2.1 Transmission 20
5.2.2 Ranging with TWR 21
5.3 Zone indication: Ensemble learning 21
5.4 Particle filter 21
5.4.1 Inputs 22

5.4.2 Prediction phase 23
5.4.3 Observation phase 23
5.4.4 Resampling phase 24
5.4.5 Variants 24
5.5 Experiment setup and parameter 25
5.5.1 Zone indication: concrete implementation and parameter . . . 25
5.5.2 Placement, trajectories and configuration 26

6 Experiments evaluation **29**
6.1 Indoor positioning results with low density of anchor nodes 29
6.2 Comments on low density anchor node test results 30
6.2.1 Bounding box restrictions 31
6.2.2 Failing UWB connections 31
6.3 Indoor positioning results with high density of anchor nodes 32
6.4 Comments on high density anchor node test results 33
6.4.1 Bounding box restrictions 33
6.4.2 Failing UWB connections 33

7 Conclusion and further work **35**
7.1 Conclusion 35
7.2 Further work 35

A Trajectories **37**
A.1 Figures of the trajectories used in the experiments 37

Bibliography **41**

List of Figures

3.1 Two way ranging 8
3.2 Time difference of arrival 9
3.3 Trilateration 10
3.4 UWB frequency spectrum 13

4.1 Particle filter design 15
4.2 System design 16

5.1 Zone definition 25
5.2 Anchor node positions 26
5.3 Trajectory 1 26
5.4 Trajectory 5 27

6.1 Positioning results trajectory 1 and 2 29
6.2 Positioning results trajectory 3 and 4 30
6.3 Bounding box and checkpoints for trajectory 1 31
6.4 Positioning results trajectory 5 32
6.5 Trajectory 5 with bounding box 33

A.1 Trajectory 1 37
A.2 Trajectory 2 37
A.3 Trajectory 3 38
A.4 Trajectory 4 38
A.5 Trajectory 5 39

List of Tables

5.1 Spectral occupation of predefined channels. 20

6.1 The arithmetic mean of errors in trajectory 1 to 4 (in meter). 30
6.2 The arithmetic mean of errors in trajectory 5 (in meter). 33

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 6 Deci**B**ell	
GHz	**G**iga**H**ert**Z**
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**ert**Z**
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
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 [22]. 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 [10]. 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, what 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 with an external 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 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.
- We created an extensive test scenario, where we placed several anchor nodes in a real building and collected data on complex 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 and the test bed. The 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 systems can either run on a seperate 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 [6]. In [2] 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 [4] 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 [14] provided a room-based ensemble learning technique for localization. The room detection uses averaged coordinate outputs of a k-NN estimator. Whereas in [3] 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 [1], 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 [5] 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 [9], 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 Combined Localization

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 [13] used a fingerprinting-based solution in combination with a digital compass. [2] 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 mitigrated 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 two sections.

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 seperate 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 because data of several target devices is available and can be further used on the server or in cloud applications.

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. A set of measurements, often received signal strength measurements, 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 simliar 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 the 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, the following formula 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 [16]:

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

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 enviroments. It can even be simplified by defining useful reference distances, like done in [12].

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 [28]. 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 [12].

3.3.2 Time of Flight Based Models

A totally different technique to get distance estimations are time of flight based models. The 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 [21].

To achieve higher accuracy, the communication can be extended with a final message containing the 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:

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

Respectively the time of flight for SDS-TWR:

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

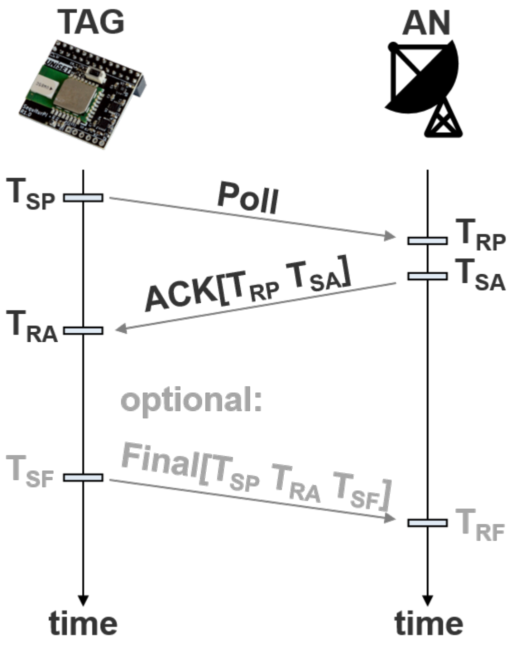

FIGURE 3.1: Illustration of TWR and SDS-TWR communication

with the following timestamps: 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 of the TAG. 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 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 [20].

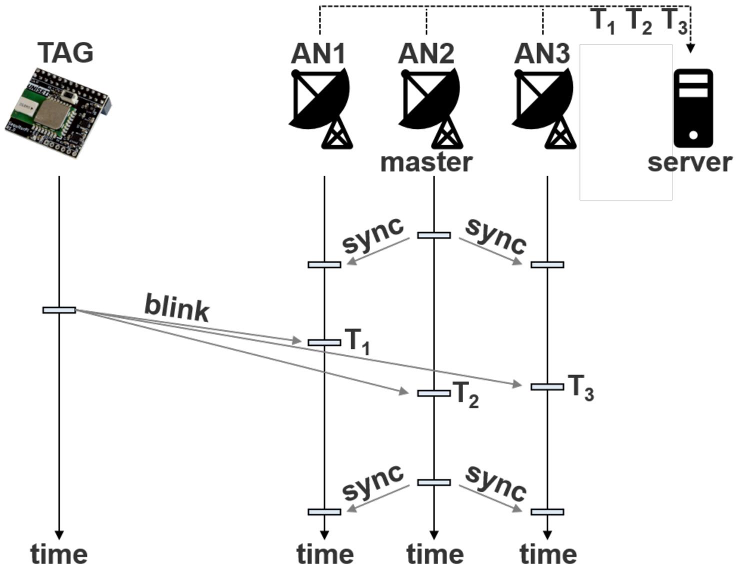

FIGURE 3.2: Typical Setup for Time Difference of Arrival communication

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 [11]. 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.

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

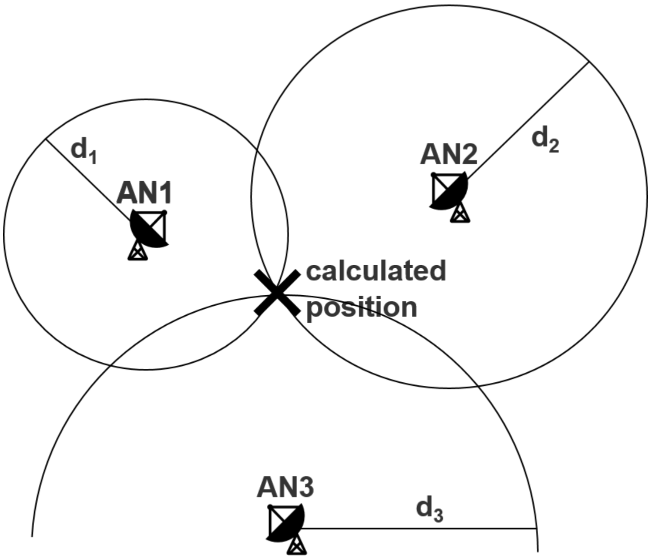

FIGURE 3.3: Graphical illustration of the trilateration concept.

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, a 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 [12] 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 an associated 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 (obstacles) 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. 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:

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

3.6 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 technologies, UWB occupies a wide area of frequencies instead of just covering a small frequency spectrum. As showed 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 communication 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.

For simplicity reasons, normally ranging systems are designed such that the TAG gets the final message of the TWR communication, as the TWR is done with several ANs. In this case the requested information - the distances to every AN - is already on one device and can be further processed. However, the TWR has not necessarily to be inizialized by the TAG - the receiver and the sender can easily be exchanged.

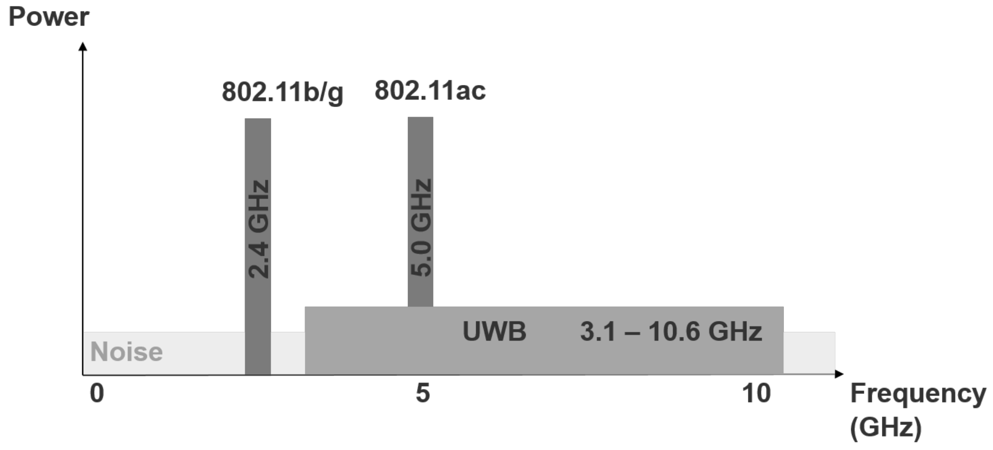

FIGURE 3.4: Comparison of Wi-Fi (802.11) and UWB frequencies.

When for example the computational power of the TAG is limited or the application runs on a separate server, we can imagine some benefits to trigger the TWR in the ANs and forward the collected data directly to the server.

Chapter 4

System design

The given background theory of section 2 was applied for our system design. In the following we present our theoretical system design. We provide a short overview in the first part, following a part where our theoretical setup is introduced. Finally we add our thoughts about the computations that are used.

4.1 Overview

As already mentioned in the introduction, our system combines different types of data input to achieve the best position estimation. This leds to a slightly more complicated ranging process than indicated on the simplified illustration in figure **??**.
We wanted to include not only measurements from one device, but from several different data sources: TAG and AN collect data and send it to the server, where the data is fed into the particle filter. The particle filter spreads particles accoring to TAG movement indication, restricted by the topology read from the floorplan and calculates the position likelihood according to range measurements, the movement vector and a rather complex zone indication. In this phase, trilateration is already done implicitly. With respect to the likelihood, a normalized weight is assigned to each particle. The position estimation corresponds to the weighted sum of all particle positions. An overview of the whole process is illustrated on figure 4.1.

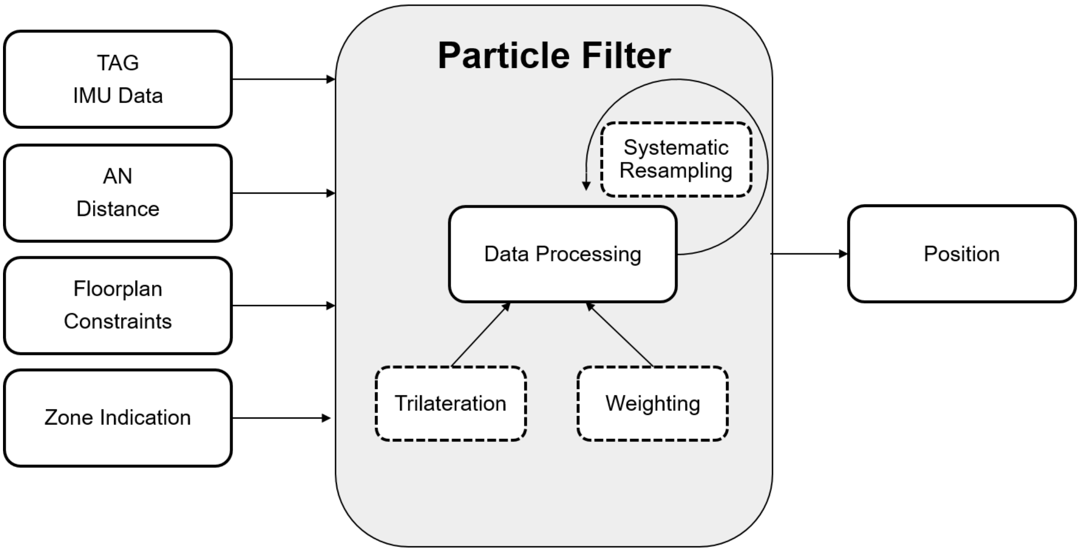
TAG
IMU Data
AN
Distance
Floorplan
Constraints
Zone Indication
Particle Filter
Systematic
Resampling
Data Processing
Trilateration
Weighting
Position

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

4.2 Setup

The theoretical design of our system is shown in figure 4.2, it works as follows: At least three ANs, equipped with UWB technology, are placed in an indoor environment of one floor. Our algorithm runs on a seperate server, where a zoneplan and floorplan of the floor are available. A single TAG is located somewhere on the floor, it is as well equipped with UWB technology, moreover it measures acceleration with the onboard IMU and signal strength to WiFi access points. The TAG continously collects data from the IMU and contemporary waits for a request from the server. The server periodically requests data via UWB from the TAG. As soon as a request is noticed at the TAG, it performs a range estimation to every AN, senses the signal strength of the WiFi and sends this data together with the continously collected IMU data to the server. For every estimation period, the server has all the data needed to feed the particle filter.

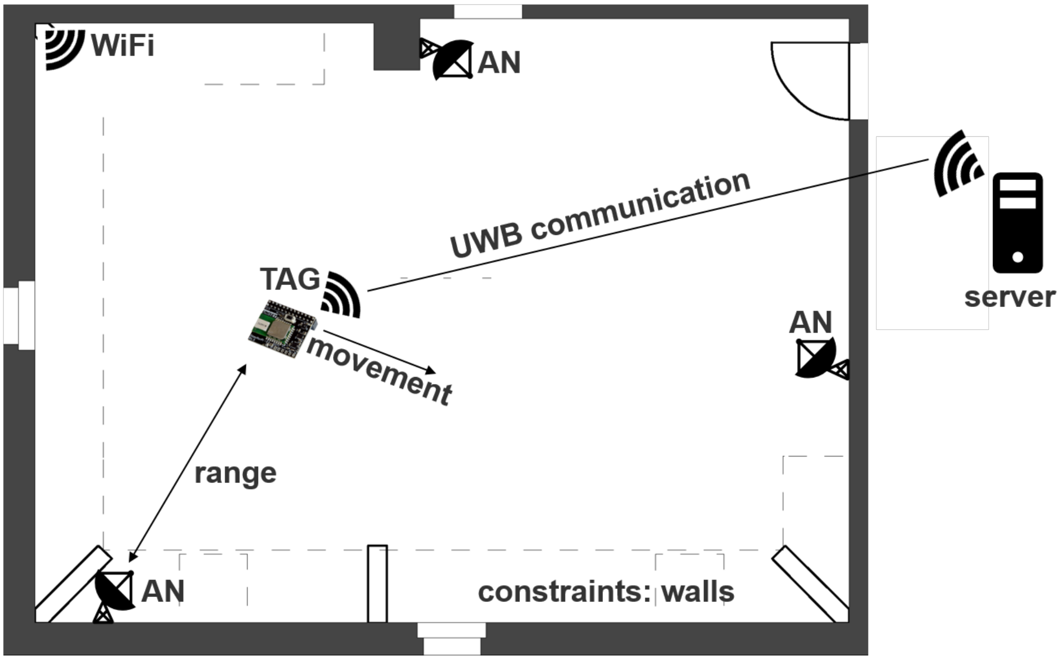

FIGURE 4.2: Overview of the theoretical system design.

4.3 Computations on the server

Most computations are done on the seperate server, as we would like to minimize the required computational power and energy consumption of the devices. As already mentioned, the server receives IMU data and range estimations from the nodes. This data, as well as the floorplan constraints and the zone indication, flow into the particle filter. These operations are done on the server:

- Spread the particles and validate new positions with floormap constraints
- Evaluate UWB ranges, IMU measures and machine learning fingerprints for zone indication to assign likelihood

- Calculate weight function and systematically resample (and reposition) particles with low weights
- Sum up weighted positions

In the first part, every particle is moved from its current to a new position. To validate the new position, we check if the direct trajectory between the old and the new location intersects with an impediment. If so, a new position is generated, as the last was not reachable. The second item covers the likelihood of the new position. For every AN, the measured distance is compared to the distance from the new position to the AN. The less these two distances differ, the higher is the assigned liklihood. In this phase also the zone indication is taken into account. A particle that lies in the predicted zone is assumed to be more accurate than particles in a wrong zone. Therefore the probability returned from the ML algorithms of being in the zone of the particle is directly assigned as the zone indication likelihood. For the IMU motion, the difference between the old and the new position is compared to the measured IMU motion to evaluate the likelihood. In the weighting step, every particle gets weighted by the liklihoods of the previous computations. The weights of all particles are normalized for further processing. The last part is simply a weighted sum of the particles locations to retrieve the estimated position.

Chapter 5

Implementation and experiment setup

In this section we explain in detail, what hardware we used in our implementation and with which technology the UWB communication and ranging was done. Then we introduce the zone indication learning algorithm and finally we briefly explain the core of our work, the mathematical theory of the particle filter.

5.1 Hardware

We decided to use Raspberry Pis for all mobile devices such as the TAG and the ANs. As the Raspberry Pis were not equipped with UWB technology, we extended them with a SEQUITUR Pi board from UNISET company running InGPS Lite firmware. In the following two subsections we present you these two hardware components.

5.1.1 Raspberry Pi

A Raspberry Pi is a single-board computer not much bigger than a credit card. Raspberry Pis are mainly designed for educational purposes as an alternative to expensive notebooks or desk computers. Hence the focus lies also on easy-to-use and plug-and-play experiences. Raspberry Pis are useful for versatile types of projects, as they provide common state of the art hardware - like HDMI, USB and wireless LAN - direct on board and as they are extendable with selected components.

We used Raspberry Pi Model B [15], these were the most relevant specifications for our work:

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN
- 100 Base Ethernet
- 40-pin extended general purpose input output (GPIO)
- Micro SD port for loading your operating system and storing data

5.1.2 SEQUITUR Pi board with InGPS Lite

On the 40-pin extended GPIO, we connected the SEQUITUR Pi board from UNISET Company. UNISET is a company located in Italy that focuses on research, development and manufacturing of innovative sensors in two major application areas [23]:

TABLE 5.1: 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

- Access control security systems, enhancing the reliability of intrusion detection
- Indoor and outdoor tracking. Sequitur is a precise real time locating system (RTLS) for tracking any object in 2D or 3D with centimeter accuracy.

This hardware seemed perfect for our ambitions, as it provides a state-of-the-art UWB communication and ranging. Moreover the SEQUITUR Pi board of the TAG has IMU sensors like 3D-accelerometer and 3D-magnetometer on board. Together with the hardware, UNISET delivers a firmware running on Raspberry Pis operating system (OS) to establish a connection via user datagram protocol (UDP). This firmware allows to communicate with the sensors in order to retrieve IMU sensor data, but also to get direct access to the range between two nodes. It is explained in detail throughout the next section.

5.2 UWB communication and ranging

The radio module of SEQUITUR Pi board is used on the one hand to transmit data - in order to obviate the need for additional communication hardware - and on the other hand to evaluate the ToF. As UNISET is a commercial company, they do not provide full information of the underlying transmission techniques. Nonetheless in the two following subsections, the known parts are mentioned.

5.2.1 Transmission

SEQUITUR InGPS Lite enables single-hop wireless communication with the UWB interface between neighboring nodes of the same network. The radio module supports six 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 5.1.

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 to communicate correctly. A lower data rate allows lager operating distances between the nodes. 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 [19] [18].

5.2.2 Ranging with TWR

UNISET company offers two different packages for positioning. InGPS Lite, which is the standard software and InGPS Pro, which is the advanced package. For our implementation InGPS Lite was sufficient, as the main difference of the two packages are the number of TAGs and anchors supported. With InGPS Lite only one TAG and a maximum of 10 anchors are supported as with InGPS Pro numerous TAGs and anchors are possible. InGPS Lite opperates only in TWR mode other than InGPS Pro, where a second mode with TDOA range estimation is available. The range estimation of two nodes is 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 client starts the TWR conversation via UWB between AN and TAG. The AN sends only one ranging request to the TAG, which immediately responds. 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 will directly determine the RTT and thus the range. Finally an answer message with the range is reported from the AN to the client and no messages are reported from the TAG to the client.

5.3 Zone indication: Ensemble learning

The zone indication is a rather complex and computationally demanding process, thus we will expain it to higher detail than the other inputs of the particle filter. The zone indication fuses Wi-Fi and UWB RSS in an enhanced learning model. A set of independent individual machine learning methods are combined in an ensemble learning model, however, we used a much easier version of the ensemble learning presented in an earlier work of CDS [3].

In our zone indication several different machine learning algorithms are fed with the same fingerprint data. Every of the machine learning algorithms performs a zone prediction and assigns a likelihood for every zone. The likelihood represents the probability of observing the given RSS fingerprint while being in this zone. Therefore the probability can be stated as: $p(y_t|z^i)$, with y_t being the observed fingerprint at time t and z^i being the i-th zone. We assume that for every of the machine learning algorithms these probabilities are statistically independent, thus the probabilities returned by our ensemble learning algorithm are just the multiplied results of the single machine learning algorithms:

$$p_{tot}(y_t|z^i) = \prod_{j=1}^{N} p_j(y_t|z^i)$$

for N as the number of descrete ML algorithms.

5.4 Particle filter

We used a particle filter approach to solve the localization problem. This method, also known as Monte Carlo Localization (MCL), is often used for indoor positioning. It combines various noisy measurements to estimate the system state and minimize errors. To introduce the particle filter, we explain in a first paragraph which inputs we fused into the particle filter. The following paragraphs define the different phases in our mathematical model whereas we discuss the different variations of our system in the last subsection.

5.4.1 Inputs

As stated above, various measurements are taken into account in our particle filter. The most important inputs are the range estimations between the TAG and ANs, the motion vector measured by the IMU of the TAG and the restrictions given by the floormap. We will refer to Zd_t as the range observation vector at time t, which is described as $Zd_t = [d_t^j], j = 1...N$, where N is the number of ANs. Every distance measurement d_t^j itself is consisting of various errors, statistically it can be described as:

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

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

The motion vector $Mv_t = [\theta_t, \ell_t]$ is modeled by the heading direction θ and the movement length ℓ. Both of θ and ℓ are calculated by the IMU readings, where again several noises occur such that the heading direction is statistically described as:

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

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. In our implementation we calculated the heading direction with the formula $\theta = atan(\frac{mag_x}{mag_y})$, where mag_x and mag_y are low pass filtered magnetometer readings in x and y direction. The update frequency of the sensor was higher than the update frequency of the particle filter, hence we calculated an average of these different measurements, hereafter appearing as θ_t for the average during time period from t-1 to t.

Whereas the heading direction is directly calculated from IMU data, for the stride length we took the previous system state into account. This was necessary, because 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. As we can not assume the acceleration to be constant, the movement length approximation can be defined as:

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

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. In our implementation the observed movement length is calculated as follows:

$$\ell_t = v_{t-1} * \Delta t + \sum_{i=0}^{N}[(a_i * \Delta t_i)(N - i)] * \Delta t$$

where v_{t-1} is the velocity of the estimated position change in the last system state update, a_i is the i-th acceleration measurement and N the number of descrete acceleration measurements during the time period Δt, which corresponds to the time passed between t-1 and t. As the acceleration sensor - depending on 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 and one with a low adaption, and only took their difference into account. This corrects a part of the long term sensor bias.

Although there were many sources of errors, for the likelihood calculation in our work we nevertheless used the actual obtained values d_t^j, θ_t and ℓ_t since the errors are handled in the likelihood model by fusing different data sources and anchor node distances. However, to compensate the bias and error for the particle spreading, we assumed the heading direction θ and the stride length ℓ as random normal variables whose values are obtained from $\mathcal{N}(\theta_t, \sigma_\theta^2)$ and $\mathcal{N}(\ell_t, \sigma_\ell^2)$.

5.4.2 Prediction phase

Each particle has a state vector that is defined as follows:

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

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. In the prediction phase each particle is updated depending of the current movement vector $Mv_t = [\theta_t, \ell_t]$. The coordinates of the particle are updated with the following pattern:

$$[x_t, y_t] = [x_{t-1} + \ell_t * cos(\theta_t), \quad y_{t-1} + \ell_t * sin(\theta_t)]$$

As mentioned in the last subsection, with ℓ_t and θ_t as random normal variables. In the remainder of this work we will also refer to the motion in Cartesian coordinates as $M_{x,t} = \ell_t * cos(\theta_t)$ for the motion in x-direction and $M_{y,t} = \ell_t * sin(\theta_t)$ for the motion in y-direction. Floorplan restrictions are applied in this phase, whereas movements through walls are not permitted, they lead to another prediction iteration for that particle.

5.4.3 Observation phase

In the observation phase an associated weight w_t^i is recalculated for every particle, since the weight does not anymore correspond to the current position. 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. Then, the probability is determined as:

$$p(Zd_t|X_t^i) = p(d_t^j|X_t^i)$$

and

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

In addition, as defined in subsection 4.3, the zone probability is:

$$p(y_t|X_t^i) = p_{tot}(y_t|z_t^i)$$

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:

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

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:

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

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:

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

5.4.4 Resampling phase

The resampling phase is an essential component of our 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 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.

5.4.5 Variants

We implemented two variants of the particle filter localization system to state the effect on accuracy of the different parts in our system. Hereafter we will refer to

PF_{full} for the full implementation of the particle filter as defined in the last subsection. However, the particles were not spread according to the movement vector, but randomly spread in a box of 2 × 2 meters centered on the last estimated position. We decided to cancel the movement vector for spreading, as it caused a very bouncy position estimation. The second variant $PF_{UWBonly}$ is exclusively using UWB ranging and does not take the IMU measured movement or the fingerprint into account, neither in the prediction phase nor in the observation phase.

5.5 Experiment setup and parameter

In our experiment we tested the localization accuracy of the different implemented variants, as well as the accuracy of the indoor tracking system UNISET company provided with their sensors, in a complex indoor scenario with trajectories through numerous of rooms on one floor in a real building of the University of Bern. During our experiments an additional test setup with optimized anchor node positions was defined.

5.5.1 Zone indication: concrete implementation and parameter

In our implementation these three completely different ML algorithms were used: Decision Tree Learning with gini impurity [25], K nearest neighbor (KNN) classification [8] and a soft voting classifier using gini, KNN and gaussian naive bayes [17, 7] algorithms. Where present, the internal parameter of the ML algorithms were optimized from training data. The three algorithms were selected out of half a dozen machine learning algorithms, those three had the best classification results in the test data.

The zone definition can be seen in figure 5.1, wheras for the smaller scenario (only for trajectory 5) only data in the passed rooms was collected. These were the rooms 1, 2, 3, 5, 6 and 7. For the wider scenario obviously all indicated rooms were fingerprinted.

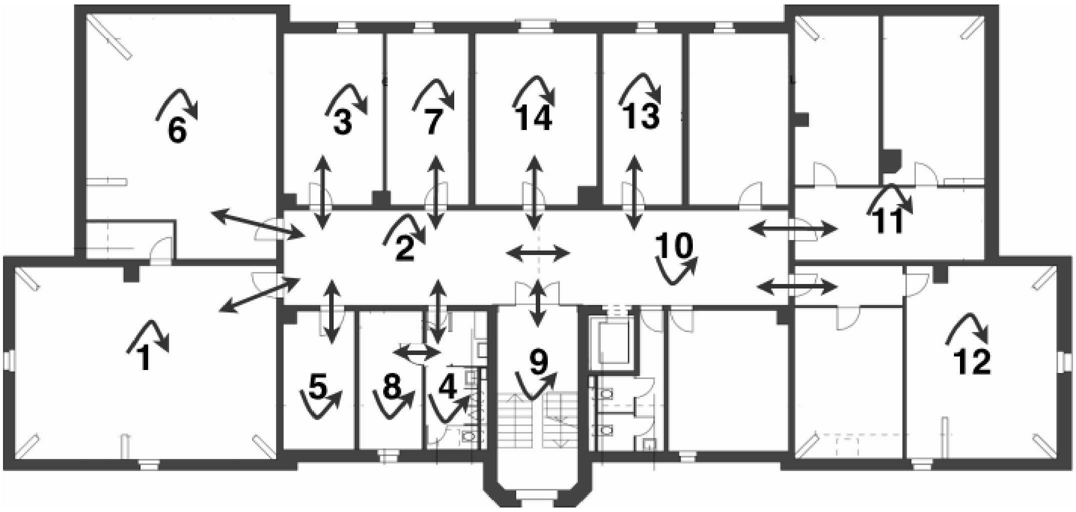

FIGURE 5.1: Zone definition and transitions between zones

For the testing data the three machine learning algorithms had a correct prediction rate of 80 to 99 percent.

5.5.2 Placement, trajectories and configuration

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 5.2. Whereas the particle filter updated its state every 700 milliseconds, the IMU sensors were updated every 100 milliseconds. The UWB transmitter of the TAG and the ANs 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. Before starting the experiments, a calibration with 1000 measurements per UWB device was made as described in the beginners guide [18].

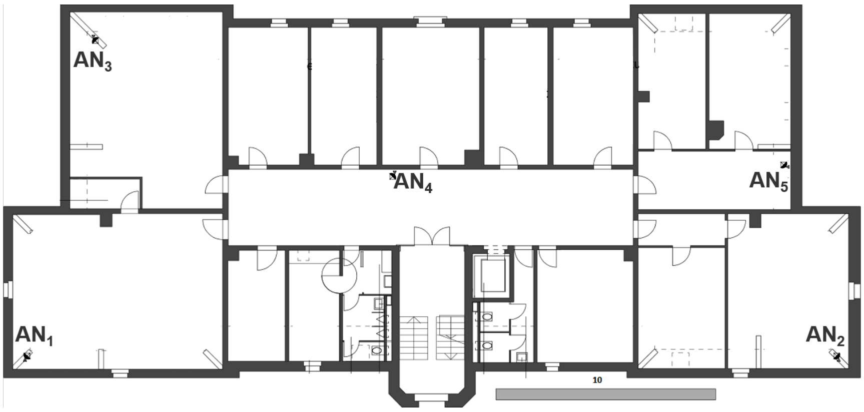

FIGURE 5.2: Distributed anchor nodes on the floor map (with distance reference of 10m).

The TAG 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 path indicated in figure 5.3, as soon as he passed a predefined checkpoint the current position estimation was registered. The other three trajectories can be seen in appendix A. Every trajectory was tested with every algorithm at least 5 times.

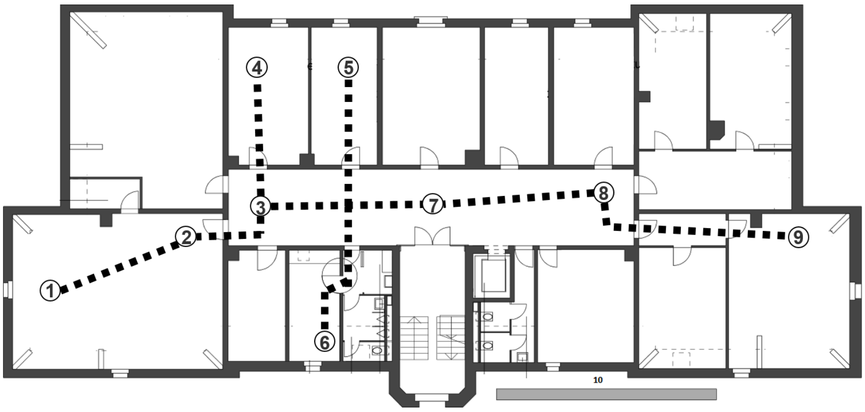

FIGURE 5.3: Trajectory 1 of the four defined trajectories with the position checkpoints.

In an ajusted experiment setup with improved anchor node positions we evaluated a fifth trajectory for every of the three algorithms. We just replaced the AN positions and did not change any other parameter. The new positions of the anchors and the trajectory checkpoints can be seen in figure 5.4. We put the anchor nodes together, such that the distance between the TAG and the farest AN was never more than 20m. Moreover, for a better comparison between the scenarios, we tried to build a similar distribution of the checkpoints as in the other trajectories. This means we added checkpoints in rooms without an AN and checkpoints at the edges of our test environment.

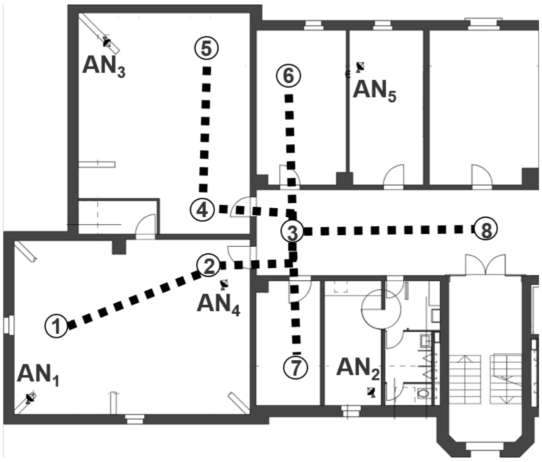

FIGURE 5.4: Trajectory 5 with improved anchor positions.

Chapter 6

Experiments evaluation

In this section we present the results of our experiments in detail. In two sections we compare the estimated positions of the three tested algorithms at the given checkpoints. Firstly the results of experiments with wide spread anchors are indicated and commented, secondly the results with nearer anchor positions are shown and discussed.

6.1 Indoor positioning results with low density of anchor nodes

In the following the results of our algorithms with a wide distance between anchor nodes are shown. Within this setup we tested trajectory 1 to 4. Each trajectory was tested five times, the results are shown as arithmetic means of these five probes. The anchor node positions for these four tested trajectories are indicated in figure 5.2.

In the first trajectory, the arithmetic mean (hereafter often called average) distance error over all checkpoints was 1.53 meter for PF_{full}, 1.47 meter for $PF_{UWBonly}$ and 1.69 meter for Sequiturs commercial system. Looking at figure 6.1, we see that the errors are often smaller than 1.5 meter, however, there are some checkpoints with very low accuracy. This is also emphasized by comparing the arithmetic mean error to the median error of 1.22m for PF_{full}, 0.68m for $PF_{UWBonly}$ and 1.14m for Sequitur, which are significantly lower than the arithmetic means.

The measurements for trajectory two were almost identical, except for $PF_{UWBonly}$, which had no outliers during trajectory 2. The arithmetic mean error for PF_{full} was 2.36m, for $PF_{UWBonly}$ it was 0.55m and for Sequitur 2.26m. Except for $PF_{UWBonly}$, the median errors were again a lot more accurate with 1.07m for PF_{full}, 0.49m for $PF_{UWBonly}$ and 1.73m for Sequitur.

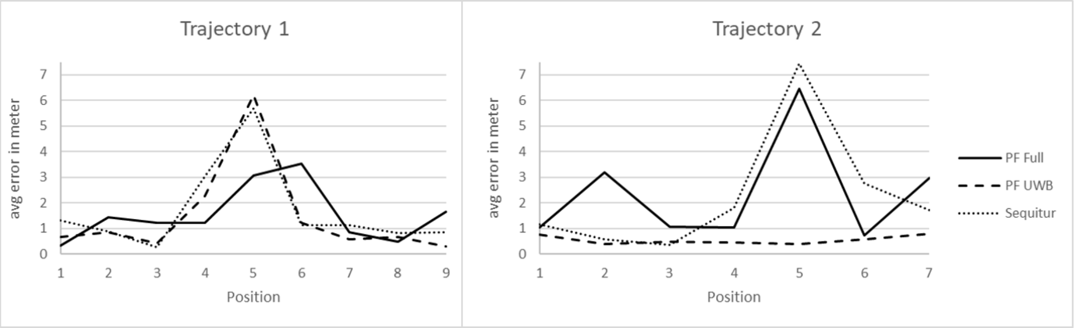

FIGURE 6.1: Graphs of measured distance errors at each checkpoint in trajectory 1, respectively trajectory 2.

The results for trajectory 3 and 4 were rather similar, however, the peaks observed in figure 6.2 were not as extreme as for the first two trajectories. As above,

TABLE 6.1: The arithmetic mean of errors in trajectory 1 to 4 (in meter).

Trajectory	PF Full	PF UWB	Sequitur
T1	1.54	1.47	1.69
T2	2.36	0.55	2.26
T3	1.18	0.80	1.87
T4	1.72	1.51	1.52
total 1-4	**1.61**	**1.03**	**1.79**

the average errors in the third trajectory of 1.18m for PF_{full}, 0.80m for $PF_{UWBonly}$ and 1.87m for Sequitur were also mentionable higher than the medians of 0.92m, 0.76m and 1.60m.

In the last of these four trajectories no big outliers were stated. Nonetheless the average errors of 1.72m, 1.51m and 1.52m, as well as the median errors 1.26m, 0.68m and 1.24m for PF_{full}, $PF_{UWBonly}$ and Sequitur, were still not as accurate as intended.

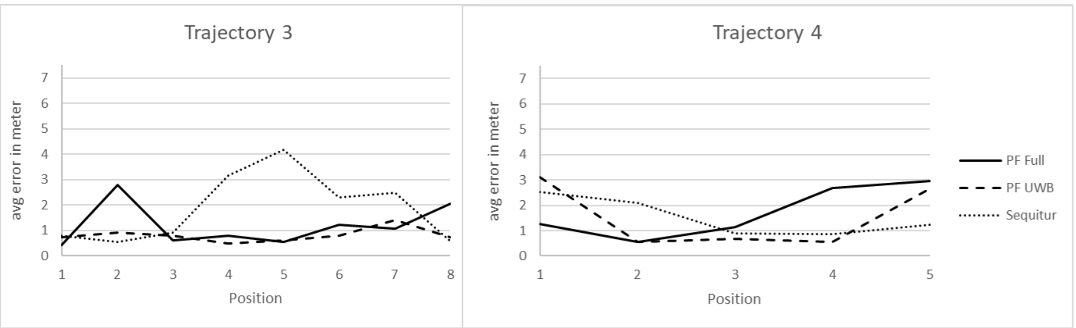

FIGURE 6.2: Graphs of measured distance errors at each checkpoint in trajectory 3, respectively trajectory 4.

Having a closer look at table 6.1, we see that surprisingly the version of the particle filter only using UWB was the best on every of the trajectories in the low-density anchor node test environment. Moreover, we have seen that there are very big differences between the single checkpoints and even between trajectories. The full variant of 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 section 6.2.

6.2 Comments on low density anchor node 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, they are explained in detail during the next two subsections.

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

These two reasons weren identified in addition to the distractions listed in chapter 2. The distractions mentioned for RSSI do also affect UWB signals, especially a cluttered environment will increase - and thus distort - 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.

6.2.1 Bounding box restrictions

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 6.3. 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_{full}, 6.23m for $PF_{UWBonly}$ and 5.70m for Sequitur during the first trajectory and 6.46m, 0.38m and 7.45m during the second trajectory are way higher than for other checkpoints. The fact that all of the three 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 far away from the nearest ANs without a direct line of sight.

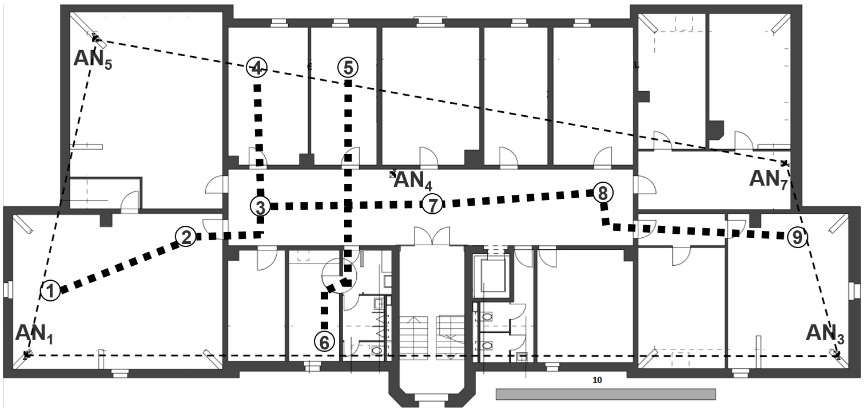

FIGURE 6.3: Indicated bounding box of the anchor node positions for trajectory 1.

6.2.2 Failing UWB connections

All three algorithms depend on a good established UWB connection, as all the ANs ranging estimations are only transmitted via UWB. Especially in the full implementation of the particle filter many UWB meassages are transmitted, because fingerprinting and IMU data are additionally requested and exchanged. For this purpose the radio mode of the UWB devices had to be set to 2, which led to a higher datarate with the cost of lower possible communication distances.

For certain checkpoints the distance to the farest anchor nodes were even more than 35m, which was too much for a stable UWB connection with the mentioned configuration. Particularly, the full particle filter had troubles receiving enough usable data, as it produced more overhead than the other algorithms. This led to a high package loss for the full version, 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.

During the experiments, we were surprised by the big inaccuracies of all three systems. We intended our system to perform better and in particular the results for Sequitur were surprising, as the UNISET company advertises with centimeter accuracy. This prompted a 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.

6.3 Indoor positioning results with high density of anchor nodes

For trajectory 5 the distances between the anchor nodes were shortened. The exact setup was indicated above in figure 5.4. Equally as before, also trajectory 5 was tested five times, the following results are shown as arithmetic means of these five tests.

As desired the results in this setup were way better than with wider ANs. With an average error of 0.45m for PF_{full}, 0.59m for $PF_{UWBonly}$ and 0.48m for Sequitur, this setup outperformed the other trajectories with every algorithm. Even the median errors - with 0.40m, 0.57m and 0.45m - were not much different, which means that there are no huge differences over the different checkpoints. This can also be seen in figure 6.4, where most of the errors are smaller than 0.80m (please note the different axis scale, when comparing with the results for trajectory 1 to 4.

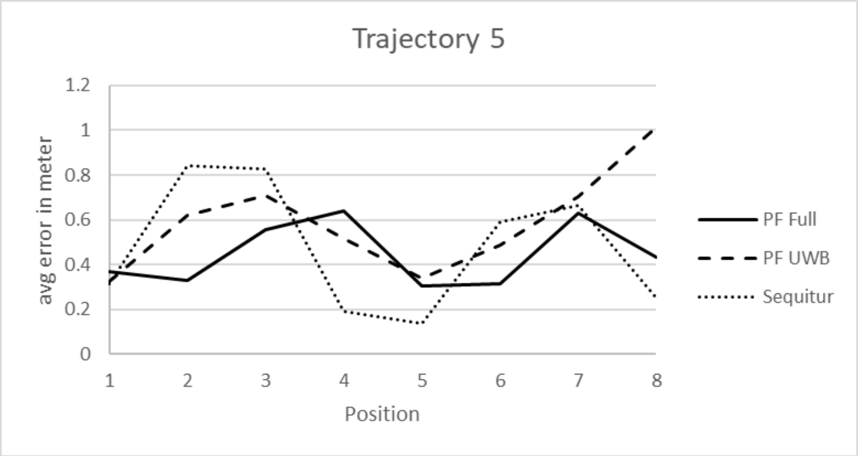

FIGURE 6.4: Graphs of measured distance errors at each checkpoint in trajectory 5.

In table 6.2, the results of the low density ANs and the high density ANs are compared. The results improved significantly, confirming our suspicion of bad communication affecting the accuracy. With the highest accuracy the full version of the particle filter slightly overcame the Sequitur commercial system. The particle filter only taking UWB ranging into account was more accurate too, however, the improvement was not as big as for the two other algorithms.

TABLE 6.2: The arithmetic mean of errors in trajectory 5 (in meter).

Trajectory	PF Full	PF UWB	Sequitur
total 1-4	1.61	1.03	1.79
T5	0.45	0.59	0.48
difference	**-1.16**	**-0.44**	**-1.31**

6.4 Comments on high density anchor node test results

With the more dense spread of the ANs, the estimations improved a lot. Having a look at our two assumptions from section 6.2, 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.

6.4.1 Bounding box restrictions

In the new setup we purposely added checkpoints at the edge or outside the bounding box, as seen in figure 6.5. 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, 0.34m, and 0.14m, for checkpoint 8 the errors were 0.43m, 1.01m and 0.25m for PF_{full}, $PF_{UWBonly}$ and 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.

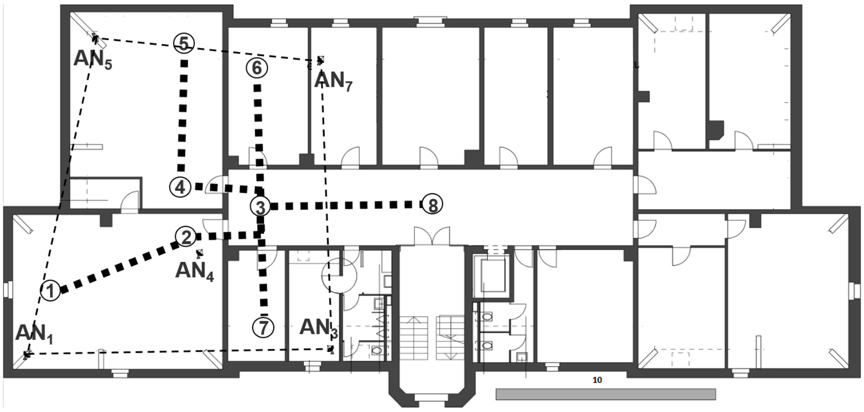

FIGURE 6.5: Indicated bounding box of the anchor node positions for trajectory 5.

6.4.2 Failing UWB connections

A bigger influence on the estimation error has 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 full version of 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 were skipped. 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.

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 0.45m 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 [2]. 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)
- Performance (run time)
- 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 performance.

For the cases with bad UWB connection, 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.

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.
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.

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

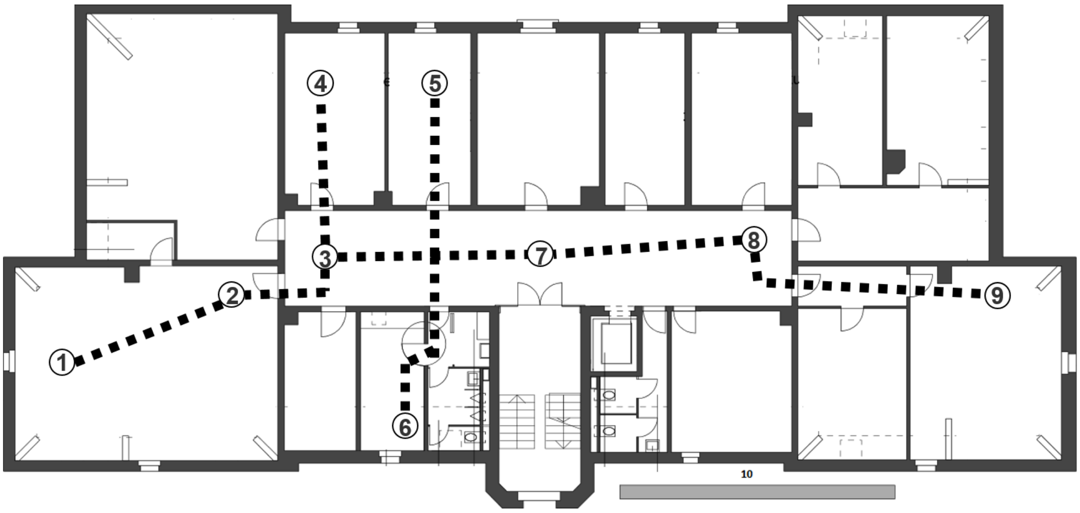
①
②
③
④
⑤
⑥
⑦
⑧
⑨
10

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

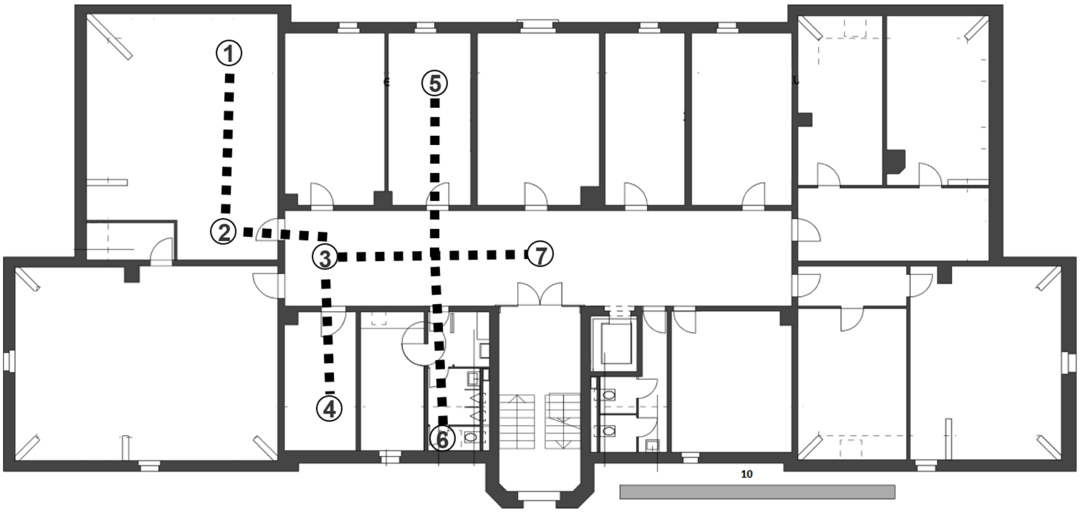

FIGURE A.2: Checkpoints and path of trajectory 2.

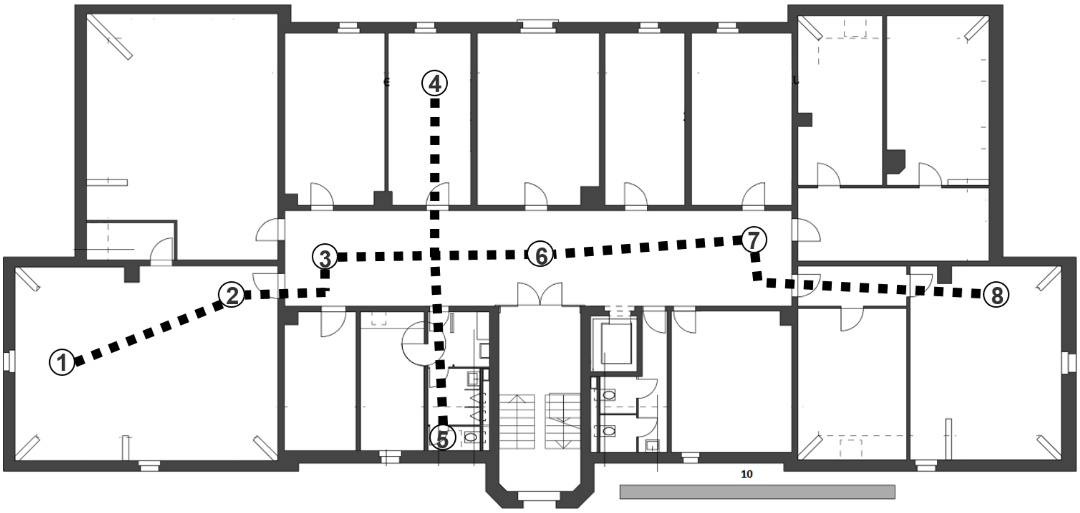
4
3
6
7
2
8
1
5
10

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

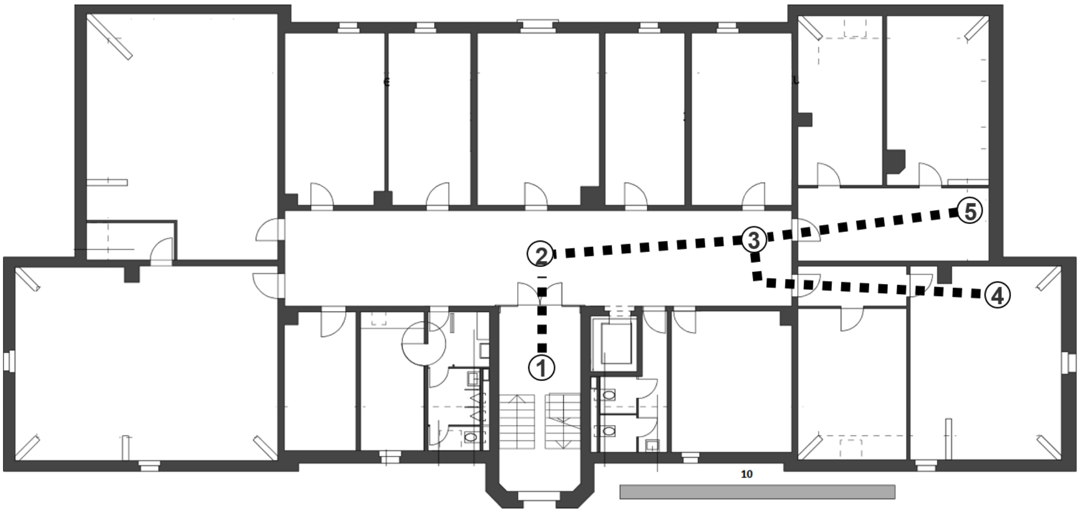

FIGURE A.4: Checkpoints and path of trajectory 4.

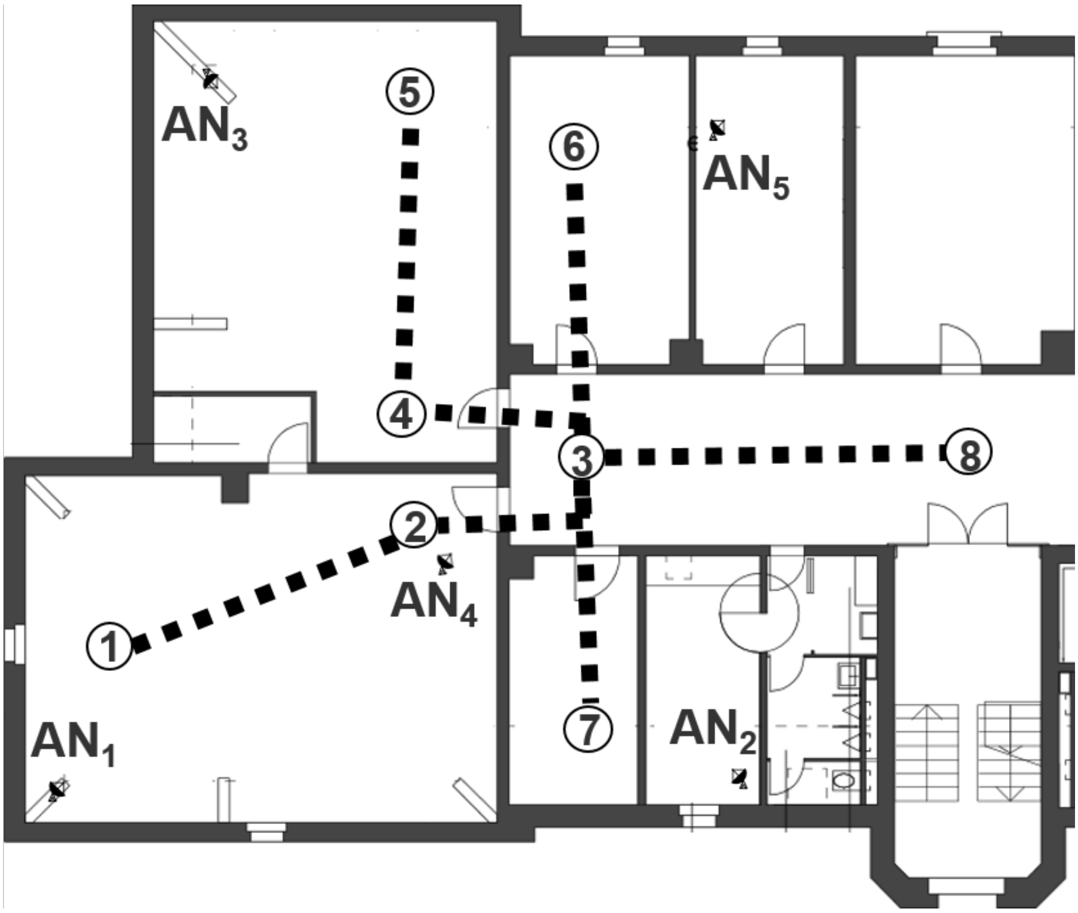
5
AN3
6
AN5
4
3
8
2
AN4
1
AN1
7
AN2

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

Bibliography

[1] ——. “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.

[2] J.L. Carrera V. et al. “A real-time robust indoor tracking system in smartphones”. In: *Computer Communications* (2017).

[3] J.L. Carrera V. et al. “Room Recognition Using Discriminative Ensemble Learning with Hidden Markov Models for Smartphones”. In: *https://arxiv.org/abs/1804.09005* (2018).

[4] Kaikai Liu et al. “Guoguo: Enabling Fine-grained Indoor Localization via Smartphone”. In: (2013).

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

[6] 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).

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

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

[9] 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.

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

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

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

[13] 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.

[14] 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.

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

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

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

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

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

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

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

[22] *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.

[23] *UNISET company: Intelligent sensor technology.* `http://www.unisetcompany.com`. Accessed: 2018-07-31.

[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] Dimitrova D.C. Zan Li Braun T. "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).