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 1, 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 Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too…

Contents

Declaration of Authorship **iii**

Abstract **v**

1 Introduction **1**
- 1.1 Motivation 1
  - 1.1.1 Indoor difficulties vs Outdoor 1
  - 1.1.2 Important Applications 1
- 1.2 Idea 2
  - 1.2.1 Ranging Positioning System with different Inputs 2
- 1.3 Contributions 2
- 1.4 Overview 3

2 Theoretical Background and Related Work **5**
- 2.1 Range based localization 5
  - 2.1.1 Received signal strength indication 6
  - 2.1.2 Round trip time: Two way ranging, time difference of arrival . 6
  - 2.1.3 Triangulation and Trilateration 9
- 2.2 UWB Theory 9
- 2.3 Particle Filter and Room Recognition 10

3 System design **11**
- 3.1 Overview 11
- 3.2 Setup 12
- 3.3 Computations on the server 12

4 Implementation and experiment setup **15**
- 4.1 Hardware 15
  - 4.1.1 Raspberry Pi 15
  - 4.1.2 SEQUITUR Pi board with InGPS Lite 15
- 4.2 UWB Communication and Ranging 16
  - 4.2.1 Transmission 16
  - 4.2.2 Ranging with TWR 16
- 4.3 Zone Indication: Descriminative Ensemble Learning with Hidden Markov Models 17
  - 4.3.1 HMM-Discriminative Ensemble Learning Method 17
  - 4.3.2 Transition and Emission Probabilities in the HMM 18
- 4.4 Particle Filter 19
  - 4.4.1 Inputs 19
  - 4.4.2 Prediction phase 21
  - 4.4.3 Observation phase 21
  - 4.4.4 Resampling phase 22
  - 4.4.5 Variants 22
- 4.5 Experiment Setup and parameter 22

4.5.1 Zone indication: concrete implementation and parameter . . . 23
4.5.2 Placement, trajectories and configuration 23

5 Experiments evaluation 27
5.1 Indoor positioning results with low density of anchor nodes 27
5.2 Comments on low density anchor node test results 28
5.2.1 Bounding box restrictions 29
5.2.2 Failing UWB connections 29
5.3 Indoor positioning results with high density of anchor nodes 30
5.3.1 Transmission 30
5.3.2 Ranging with TWR 30
5.4 Comments on high density anchor node test results 30
5.4.1 Inputs 30
5.4.2 Likelihood and weighting 30
5.4.3 Ranging with TWR 30
5.5 Conclusion and further work 30

A Trajectories 31
A.1 Figures of the trajectories used in the experiments 31

Bibliography 35

List of Figures

2.1 Ranging process 5
2.2 Two Way Ranging 7
2.3 Time Difference of Arrival 8
2.4 Trilateration 9
2.5 UWB frequency spectrum 10

3.1 Particle Filter Design 11
3.2 System Design 12

4.1 HMM 18
4.2 Zone Definition 23
4.3 Anchor Node Positions 24
4.4 Trajectory 1 24
4.5 Trajectory 5 25

5.1 Positioning results trajectory 1 and 2 27
5.2 Positioning results trajectory 3 and 4 28
5.3 Bounding box and checkpoints for trajectory 1 29

A.1 Trajectory 1 31
A.2 Trajectory 2 31
A.3 Trajectory 3 32
A.4 Trajectory 4 32
A.5 Trajectory 5 33

List of Tables

5.1 The arithmetic mean of errors in trajectory 1 to 4 and in total over all measured checkpoints for the three algorithms. 28

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
GHz	**G**iga**H**ert**Z**
GPIO	**G**eneral **P**urpose **I**nput **O**utput
GPS	**G**lobal **P**ositioning **S**ystem
HMM	**H**idden **M**arkov **M**odel
IMU	**I**nertial **M**easurement **U**nits
IoT	**I**nternet **o**f **T**hings
kbps	**k**ilo **b**it **p**er **s**econd
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
MLP	**M**ulti **L**ayer **P**erceptron
M2M	**M**achine **2**(to) **M**achine
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

1.1 Motivation

In the last twenty years, the number of mobile devices in use has tremendously increased. In the first quarter of 2018 more than 380 Million smartphones have been sold worldwide [9]. However, in the past few years, not only smartphones have been sold, but also 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 30 Billion devices in 2020 and already more than 70 Billion in 2025 [18].

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.

1.1.1 Indoor difficulties vs Outdoor

For outdoor applications, primarily the Global Positioning System (GPS) is in use. For indoor application in the other hand, GPS has limitations that make it almost useless. Due to the environmental conditions indoors, with heavy walls armoured with steel and other distractions, additional signal loss is encountered which makes it hard to detect and decode GPS signals [10]. In addition, higher buildings in the neighborhood can reflect transmitted signals, which leads to false position estimations. As GPS is mainy applied as 2D positioning system, it will not provide 3D indoor information such as the current floor level. For this purposes we are forced to use alternative technologies that provide even higher accurracy indoors than GPS would achieve outdoors. There are many different approaches to do indoor positioning, which made it an attractive and active research field.

1.1.2 Important Applications

There are various possible use cases for devices that track their indoor position. These use cases can be grouped into two groups. In the one hand applications for pedestrians with a smartphone and in the other hand real machine to machine (M2M) applications.
Some examples for smartphone applications:

Location of person in need: For emergency services every second counts to get to the position of persons in need. An accurate positioning system that indicates additional information such as the floor level could save lifes.

Security Guards: Real time tracking of security guards on their patrol. A security system can check autonomous if all security guards are on the right tracks.

Museum guidance: Tourists visiting a museum could easily be guided through the museum with customized location based information.

Examples for machine to machine (M2M) applications:

Logistic: An autonomous storage system can find articles in a big storehouse according to the exact position of the carrier vehicle. Numerous vehicles can be in use at the same time.

Cleaning: An autonomous cleaning machine keeps track of its position, such that the floor can efficiently be cleaned.

Indoor post roboter: An autonomous roboter can collect letters in the building and bring them to the internal post office.

1.2 Idea

For an object in space, there are several basic ideas to keep track of its current location. We can define a starting position and keep track of every move the device registers. E.g. every visitor in the museum starts at the entrance and will then walk through the building. Alternatively the object can be tracked by defining at least three triangulation points and periodically measure the distance from these points to the device. There are various ways to measure this distance, some with higher and some with lower accuracy.

1.2.1 Ranging Positioning System with different Inputs

Our idea was to not only use one of the mentioned approaches, but to combine them in one alorithm. We would use a range positioning system combined with motion detection of the device and even integrate environmental restrictions, given by floor topologies like walls. By combining different methods we hope to compensate measurement errors and thus minimize the overall errors.

1.3 Contributions

In this thesis we present a real-time indoor positioning system running on a Raspberry Pi, based on a particle filter implementation in smartphones that was developed in previous works of the University of Bern [2]. We adapted the inputs of the particle filter to range-based localization using ultra wideband (UWB) instead of Wi-Fi and added motions measured by inertial measurement units (IMU) of the target. In addition, we implemented a room recognition algorithm using WiFi and UWB RSS fingerprints and fused its results into our particle filter approach. We expound results of our experiments, where we tested different variants of our implementation and other algorithms in a real test scenario and compared the accuracy of the estimated position.

Our main contributions are:

- We implemented a real-time localization system on Raspberry Pi using UWB and IMU sensors, as well as a room recognition based on fingerprinting.
- We created an extensive test scenario, where we placed several anchor nodes in a real building and collected data on complex indoor trajectories.

- We compared the results of our implementation to the results of an UWB based localization system provided by Uniset Company.

1.4 Overview

Our work compounds of five remaining chapters: Section 2 provides the theoretical background and related work. Chapter 3 presents the theoretical system design and chapter 4 more specifically explains our system implementation and the test bed. The evaluation of our experiments can be found in section 5. Finally the sixth part concludes the work, where our findings are summarized.

Chapter 2

Theoretical Background and Related Work

In this section we explain the different types of range measuring in range-based localization systems. For comparison reasons, we shortly introduce variants of received signal strength indication (RSSI), which is the mostly used indoor localization technique. We then briefly explain two slightly less common methods, which were used in our system - two way ranging (TWR) and time difference of arrival (TDOA). We also include some background theory about our implementation and the particle filter.

These are the main parts of this section:

First a short overview of range based localization with the main principles of RSSI, TWR and TDOA as well as the concept of triangulation/trilateration and the weighting process. Second we present background information about ultra wideband (UWB) and finally information about the particle filter is given.

2.1 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.

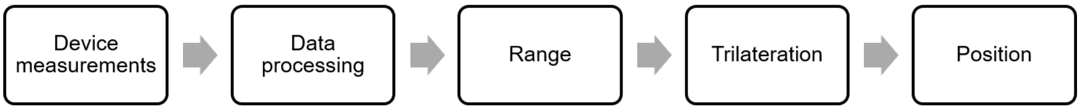

FIGURE 2.1: A simple ranging process.

As shown in figure 2.1, a simplified localization system work as follows: Either the TAG, the anchor or both of them collect data used for localization. The data can be a signal strength, a round trip time or IMU measurement. In a processing unit - on the TAG or on a seperate server - the data is processed and converted into a distance. This is repeated for every anchor node. The last step contains trilateration of the position using the ranges of every AN to the TAG.

In this abridged scenario, some difficulties are left out. Full indoor localization systems are more complex, as they use ingenious algorithms to improve the accuracy of the estimated ranges or improve the system by adding weighting to deal with incorrect range measures.

2.1.1 Received signal strength indication

As already mentioned, many indoor positioning algorithms use received signal strength indication to calculate distances to the anchor nodes. Mainly because RSSI can be applied to almost every type of transmitted signal, thus RSSI uses the universal theory of free-space path loss. The following formula describes the relation between the received signal strength and the distance to the transmitter [6].

$$P_r = P_t(\lambda/4\pi r)^2$$

P_r - received signal strength
P_t - transmitted signal strength
λ - wavelength
r - radius (distance from transmitter to receiver)

However, this formula is restricted to free-space. There are several different kind of distractions that can affect the accuracy of the measurements in an indoor environment. For example the following occurences:

- Multi-path propagation
- Reflections
- Diffraction
- Doppler effect
- etc.

Signal strength can often be obtained from the transmitter hardware as a discrete number. The higher the number, the stronger the signal. This discretization reflects another source of errors.

2.1.2 Round trip time: Two way ranging, time difference of arrival

Gathering round trip times (RTT) is a second method to get distance estimations. For accurate RTT results the hardware of transmitter and receiver, as well as the operating firmware are very important. For the presented two RTT-measuring communication techniques, the key characteristics are either a quick responding time or extremely well synchronized TAG and AN.

Operating in two way ranging (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 the equation below, the time of flight can now be evaluated [17]. To achieve higher accuracy, the communication can be extended with a final message containing the timestamps of the requester. This is called symmetrical

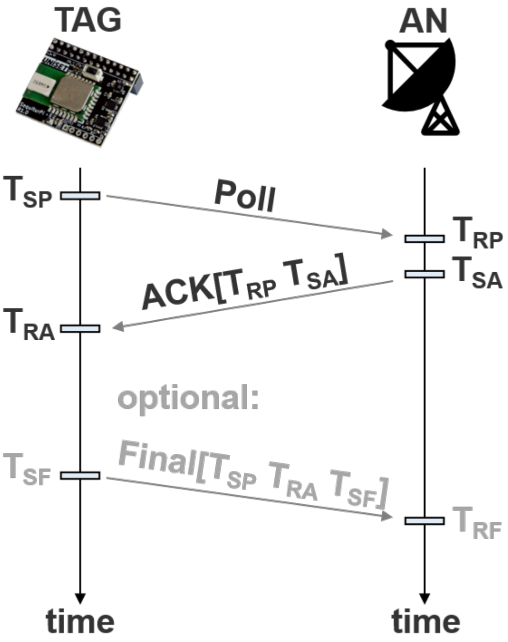

FIGURE 2.2: Illustration of TWR and SDS-TWR communication

double-sided two-way ranging (SDS-TWR) [20] and is indicated as optional in Figure 2.2.

Time of flight for TWR:

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

Time of flight for SDS-TWR:

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

ToF - Time of flight
T_{SP} - sending of poll timestamp
T_{RP} - reception of poll timestamp
T_{SA} - sending of ACK timestamp
T_{RA} - reception of ACK timestamp
T_{SF} - sending of final timestamp
T_{RF} - reception of final timestamp

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. 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 direclty to the server.

While TWR does not need further synchronization between the devices, time difference of arrival (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. 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 for a tagret device, it can perform the position and therefore the distance estimation. 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 2.3. For TWR the overhead grows enourmous with every anchor and every tag that is added. For TDOA hundrets of TAGs can be tracked, whith 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 [16].

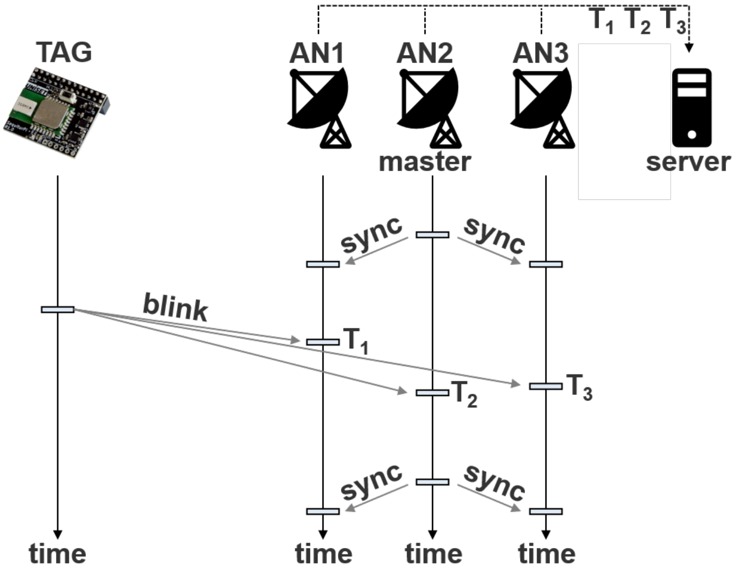

FIGURE 2.3: Typical Setup for Time Difference of Arrival communication

For both methods the same calculations to convert the ToF to the related distance can be applied. We assume that radio signals travel almost with speed of light, so we just multiply the ToF with speed of light (c_0).

$$distance = ToF * c_0$$

ToF - Time of flight

c_0 - Speed of Light = $2.997\,924\,58 \times 10^8\,\mathrm{m\,s^{-1}}$ (exact)

2.1.3 Triangulation and Trilateration

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 2.4 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.

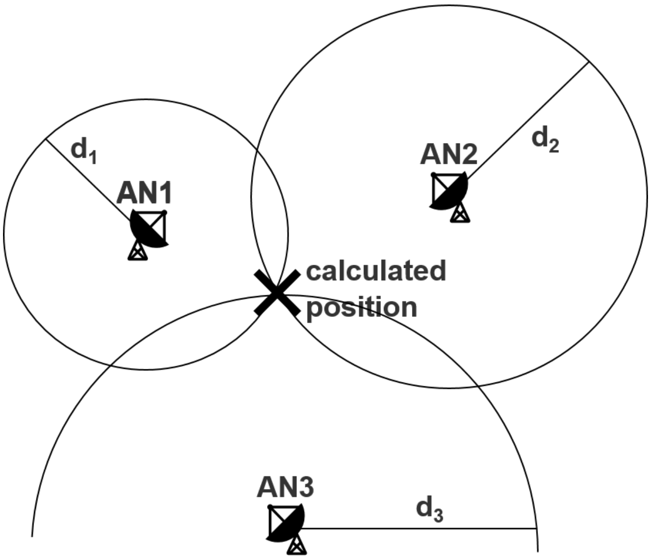

FIGURE 2.4: Graphical illustration of the trilateration concept.

2.2 UWB Theory

Ultra wide-band (UWB) is a radio technology in use for military 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 2.5, 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. Using the large frequency spectrum, there are even methods to overcome multipath propagation, when at least some frequencies have direct line-of-sight trajectories.

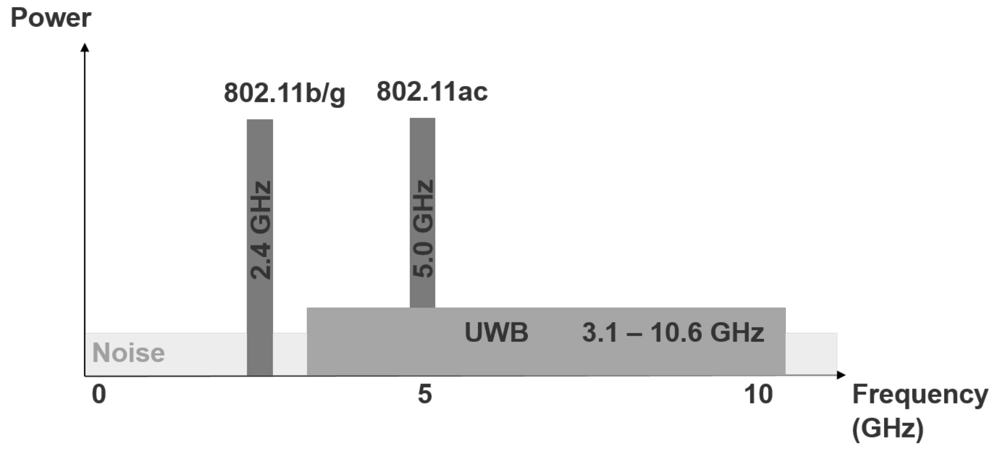

FIGURE 2.5: Comparison of Wi-Fi (802.11) and UWB frequencies.

2.3 Particle Filter and Room Recognition

In the last years the CDS of UniBe has made big effort in developing an indoor positioning system. In various works, many variants of particle filter based localization have been presented and tested. In one of the latest works, the authors proposed a robust real-time indoor tracking system based on smartphones [2], which extended some previous works [1]. These works were mainly based on Wi-Fi RSSI measurements fusing with IMU data. Since there are no documentations about the particle filter approach based on UWB ranging, we will address this gap.
In addition, the CDS also presented enhanced machine learning techniques for room recognition based on Wi-Fi and magnetic field energy fingerprints [3].

As our work is an extension of the particle filter in combination with the room recognition algorithm, the principles of these works are explained in detail in chapter 4.

Chapter 3

System design

The given background theory of section 2 was applied for our system design. In the following we present our theoretical system and which aspects of the theory led to system design decisions. 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.

3.1 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 2.1.
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 3.1.

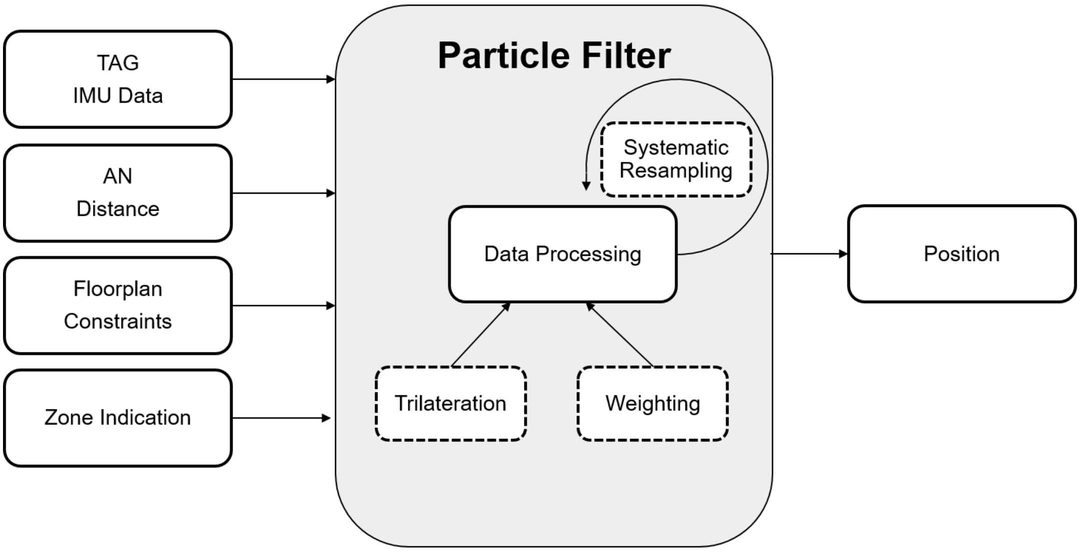

FIGURE 3.1: Overview of the theoretical components in our system.

3.2 Setup

The theoretical design of our system is shown in figure 3.2, 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.

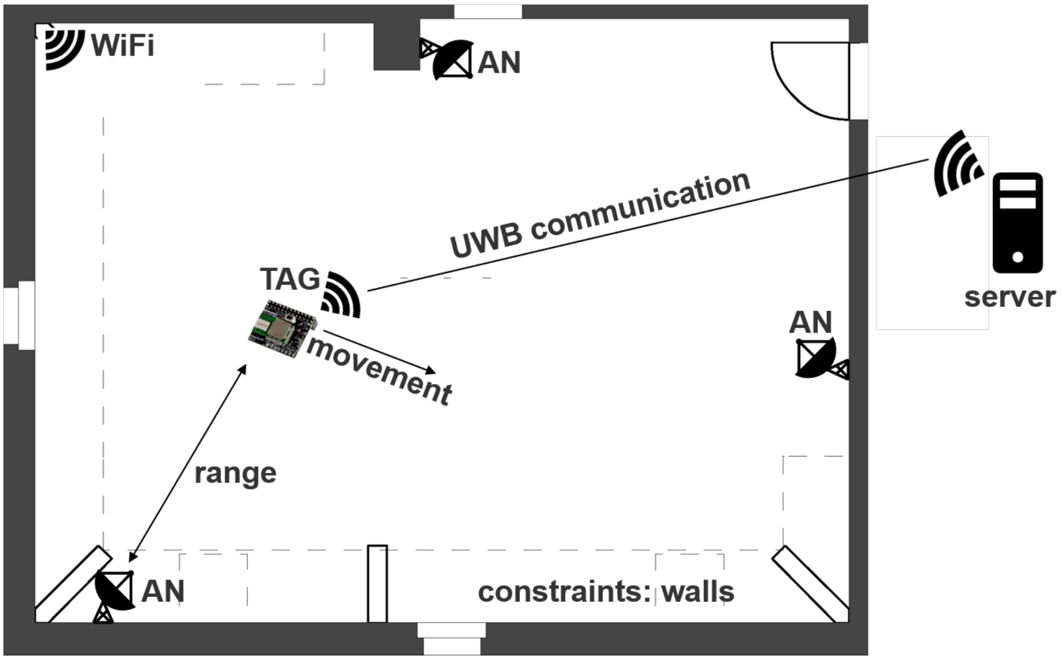

FIGURE 3.2: Overview of the theoretical system design.

3.3 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 positon 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.

Chapter 4

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.

4.1 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.

4.1.1 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 boad and as they are extendable with selected components.

We used Raspberry Pi Model B [12], 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

4.1.2 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 [19]:

- 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.

4.2 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 comercial company, they do not provide full information of the underlying transmission techniques. Nonetheless in the two following subsections, the known parts are mentioned.

4.2.1 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 the table below:

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

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 [15] [14].

4.2.2 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 2.2. 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.

4.3 Zone Indication: Descriminative Ensemble Learning with Hidden Markov Models

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, UWB, magnetic field (MF) and room transition information in an enhanced learning model. A set of independent individual machine learning methods are combined in an ensemble learning model. The used approach integrates Hidden Markov Model (HMM) with discriminative learning techniques as presented in an earlier work of CDS [3].

4.3.1 HMM-Discriminative Ensemble Learning Method

In this approach different machine learning algorithms are combined to improve the zone prediction result. A zone can be defined as any subarea in the area of interest (e.g. a room). In the concept of Markov localization [4], the state of the system is estimated by the state transitions. For localization, the states of the model correspond to defined zones. Therefore, the HMM is specified by the following components, as in [3]:

- A set of n states $Z = \{z_1, z_2, \dots, z_n\}$, with z_i as the identifier value of the zone i. Resulting the descrete random variable $x_t \in Z$ representing the hidden state at time t.
- A quadratic matrix A holding the transition probabilities,

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix},$$

  where a_{ij} respresents the probability of moving from zone z_i to zone z_j.
- A set of observations O, defined as:

$$O = \{(o_1, o_2, \dots, o_m)_1, \dots, (o_1, o_2, \dots, o_m)_r\}$$

where o_i stands for the zone prediction result of the i-th individual machine learning algorithm. This leads to O being a set of ${}^{r}P_{m}$ permutations (repetitions allowed), with r being the number of zones and m the number of different machine learning algorithms. The machine learning methods have to be conditionally independent for $y_t \in O$ as the random variable of the observations in time t.

- A matrix B holding the emission probabilities of observation likelihoods:

$$B = \begin{pmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,r} \\ b_{2,1} & b_{2,2} & \dots & b_{2,r} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \dots & b_{n,r} \end{pmatrix},$$

where b_{ij} respresents the probability of $(o_1, o_2, \dots, o_m)_j$ being the observation generated in zone z_i.

- An initial probability distribution $\pi = \pi_1, \pi_1, \dots, \pi_n$ over the n zones.

How the specified components interact in the HMM can be seen in figure 4.1.

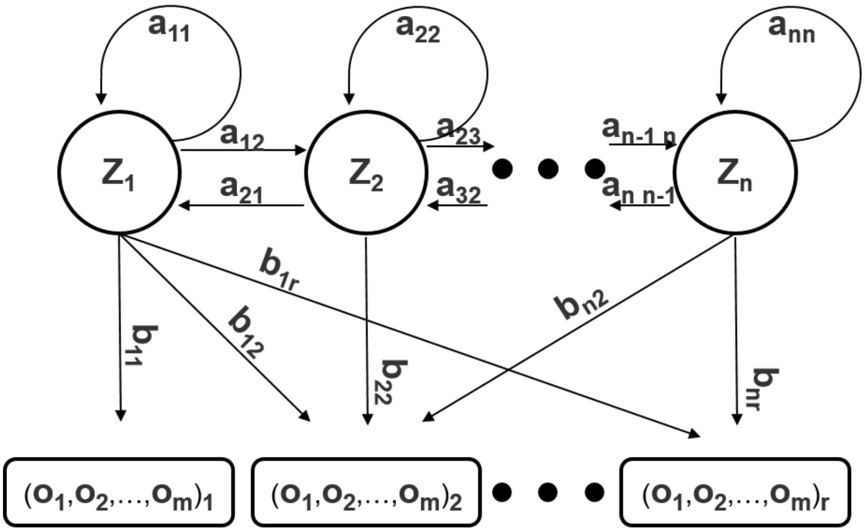

FIGURE 4.1: Application of the different probabilistic parameters in the hidden markov model.

The learning of each individual machine learning method only relies on the fingerprint of Wi-Fi, UWB and magnetic field, that's why prediction errors can still occur. However we can integrate the zone transition information (e.g. some areas are only reachable through other areas) in the HMM to improve the prediction. To determine the sequence of variables that is underlying source of some sequence of observation in a model with hidden variables, a decoding task is necessary. With the HMM model $\lambda = \{\pi, A, B\}$ and the given observation sequence $y_{t-i}, \dots, y_{t-1}, y_t$, the Viterbi algorithm [8] is used to estimate the hidden states sequence $x_{t-i}, \dots, x_{t-1}, x_t$.

4.3.2 Transition and Emission Probabilities in the HMM

In advance, all zones are exactely defined. Then, the transition probabilies in matrix A express the likelihood of moving from a beforehand defined zone to another. The connections between zones, retrieved from the floorplan, predetermine these probabilities. Therefore the values in the $n \times n$-matrix $A = (a_{ij})$ are defined as follows:

$$a_{ij} = P(x_{t+1} = z_j | x_t = z_i)$$

where a_{ij} represents the transition likelihood from zone z_i to zone z_j. Thus, $\forall i \in \{1, \ldots, n\} : \sum_{j=1}^{n} a_{ij} = 1$.
The emission probability is the likelihood of observing a particular set of observations y_j at zone z_i. Therefore the values of the $n \times r$-matrix $B = (b_{ij})$ can be written as:

$$b_{ij} = P(y_j | z_i), \forall y_j \in O \wedge z_i \in Z,$$

where $y_j = (o_1, o_2, \ldots, o_m)_j$ and o_i is the zone prediction result from the i-th machine learning algorithm. Since different individual machine learning methods are used independently, we can assume with good reason that their outcomes are conditionally independent. This leads to the simplified expression:

$$b_{ij} = \prod_{n=1}^{n} P(o_j | z_i)_n,$$

where $P(o_j | z_i)_n$ is the probability of the n-th learning method predicting o_j at zone z_i. Therefore it is equal to the sensitivity of the n-th machine learning algorithm at zone z_i:

$$P(o_j | z_i)_n = \frac{TP_n}{TP_n + FN_n},$$

with TP_n and FP_n being the true positive respectively the false positive rate of the corresponding ML algorithm.

4.4 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.

4.4.1 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 $\mathcal{N}(\ell_t, \sigma_\ell^2)$.

4.4.2 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.

4.4.3 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:

$$p(y_t|X_t^i) = p(y_t|z_t^i)$$

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:

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

4.4.4 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.

4.4.5 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.

4.5 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.

4.5.1 Zone indication: concrete implementation and parameter

In our implementation these three completely different ML algorithms were used: KStar [5], Multilayer Perceptron (MLP) [7] and the J48 decision tree [13]. **ADAPT to GINI, SV and KNN** The internal parameter of the ML algorithms were optimized from training data, whereas the following parameter were used for the non self optimizing parameters: global percent ratio of 30% for KStar, single hidden layer with 10 neurons for MLP and a confidence factor of 0.25 for J48. The zone definition can be seen in figure 4.2.

We assumed that the liklihood of staying in the same zone is bigger than moving to another zone, which led to the empirically defined values for matrix A as:

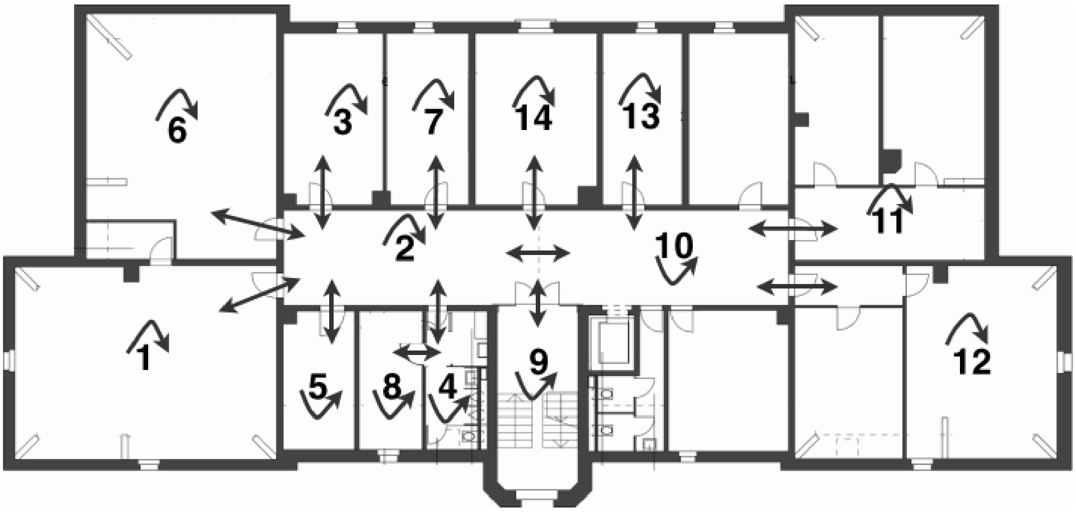

FIGURE 4.2: Zone definition and transitions between zones

MATRIX A has to be updated by the correct values

$$A = \begin{pmatrix}
0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.6 & 0.4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix},$$

4.5.2 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 4.3.

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.

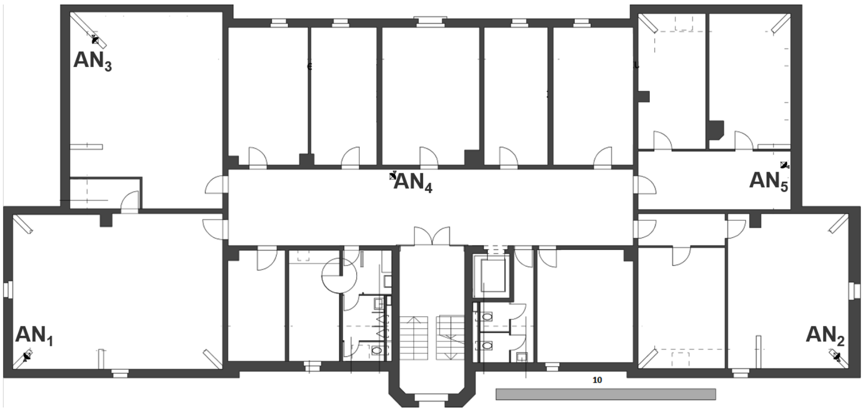

FIGURE 4.3: Distributed anchor nodes on the floor map (with distance reference of 10m).

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 4.4, 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 4 times.

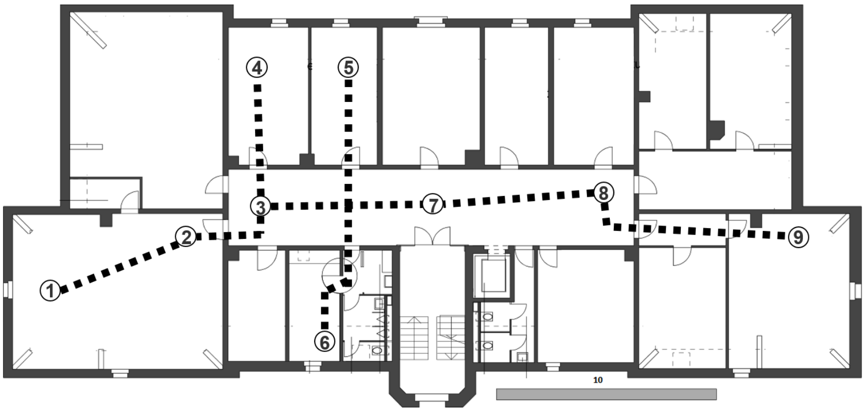

FIGURE 4.4: Trajectory 1 of the four defined trajectories with the position checkpoints.

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 4.5.

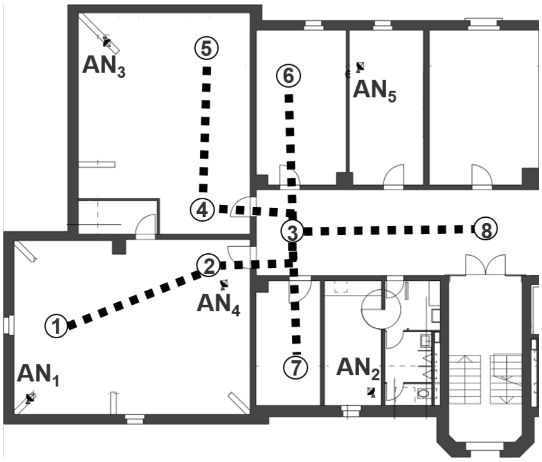
AN$_3$
⑤
⑥
AN$_5$
④
③
⑧
②
AN$_4$
①
AN$_1$
⑦
AN$_2$

FIGURE 4.5: Trajectory 5 with improved anchor positions.

Chapter 5

Experiments evaluation

In this section we present the results of our experiments in detail. In the first 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, secondly the results with nearer anchor positions are shown. In a third section we justify the differences between these two scenarios and comment the accuracy of the results. In the last section we conclude our work and propose next improvement steps for our algorithm.

5.1 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 4.3.

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 5.1, 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.

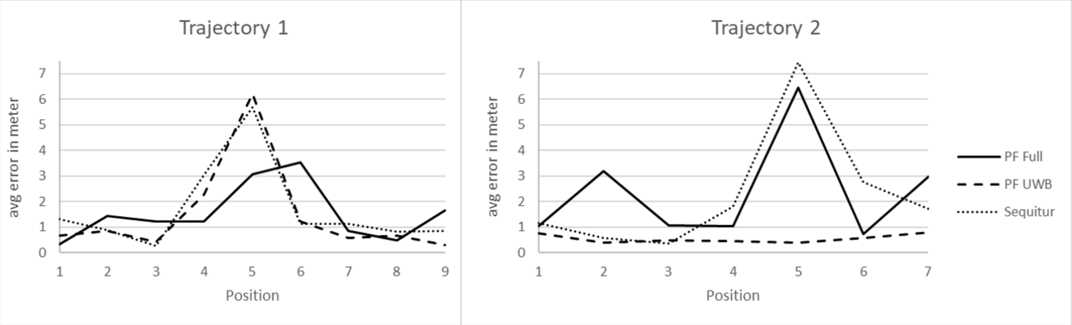

FIGURE 5.1: Graphs of measured distance errors at each checkpoint in trajectory 1, respectively trajectory 2.

TABLE 5.1: The arithmetic mean of errors in trajectory 1 to 4 and in total over all measured checkpoints for the three algorithms.

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	**1.61**	**1.03**	**1.79**

The results for trajectory 3 and 4 were rather similar, however, the peaks observed in figure 5.2 were not as extreme as for the first two trajectories. As above, 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.

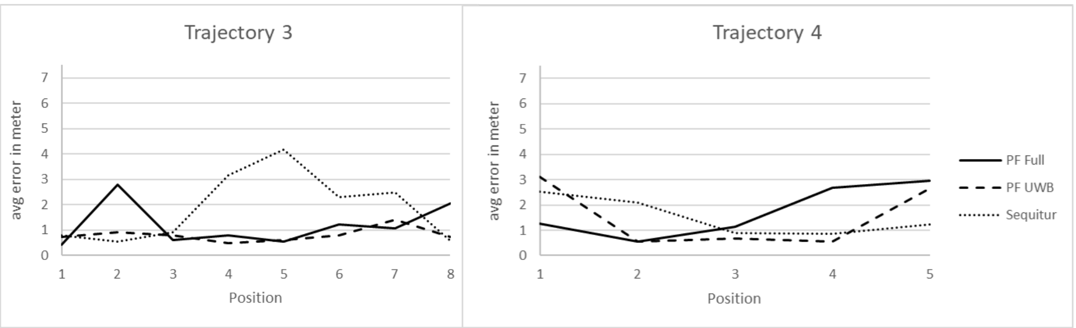

FIGURE 5.2: Graphs of measured distance errors at each checkpoint in trajectory 3, respectively trajectory 4.

Having a closer look at table 5.1, 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 5.2.

5.2 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

5.2.1 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 5.3. 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 $PF_{U}WBonly$ 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 line of sight.

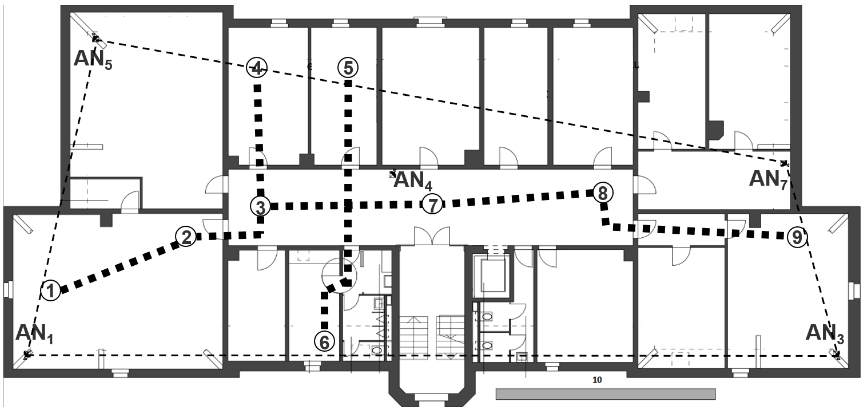

FIGURE 5.3: Indicated bounding box of the anchor node positions for trajectory 1.

5.2.2 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 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.

5.3 Indoor positioning results with high density of anchor nodes

By

5.3.1 Transmission

Here goes the text.

5.3.2 Ranging with TWR

Here goes the text.

5.4 Comments on high density anchor node test results

Here goes the text.

- Spread the particles and validate new positions with floormap constraints
- Evaluate UWB ranges, IMU measures and zone indication to assign likelihood
- Calculate weight function and systematically resample (and reposition) particles with low weights
- Sum up weighted positions

5.4.1 Inputs

Here goes the text.

5.4.2 Likelihood and weighting

Here goes the text.

5.4.3 Ranging with TWR

Here goes the text.

5.5 Conclusion and further work

Here goes the text.

- Spread the particles and validate new positions with floormap constraints
- Evaluate UWB ranges, IMU measures and zone indication to assign likelihood
- Calculate weight function and systematically resample (and reposition) particles with low weights
- Sum up weighted positions

Appendix A

Trajectories

A.1 Figures of the trajectories used in the experiments

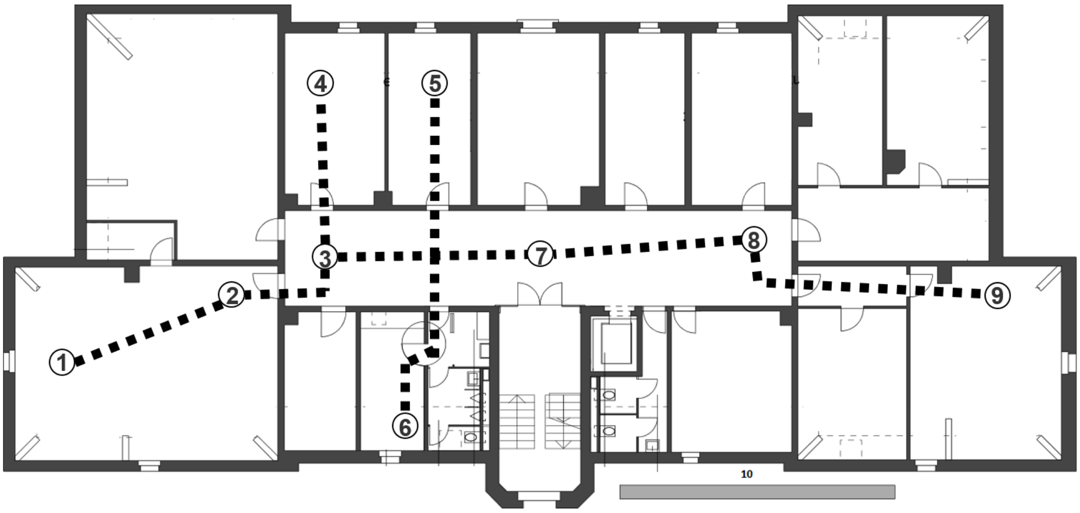
①
②
③
④
⑤
⑥
⑦
⑧
⑨
10

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

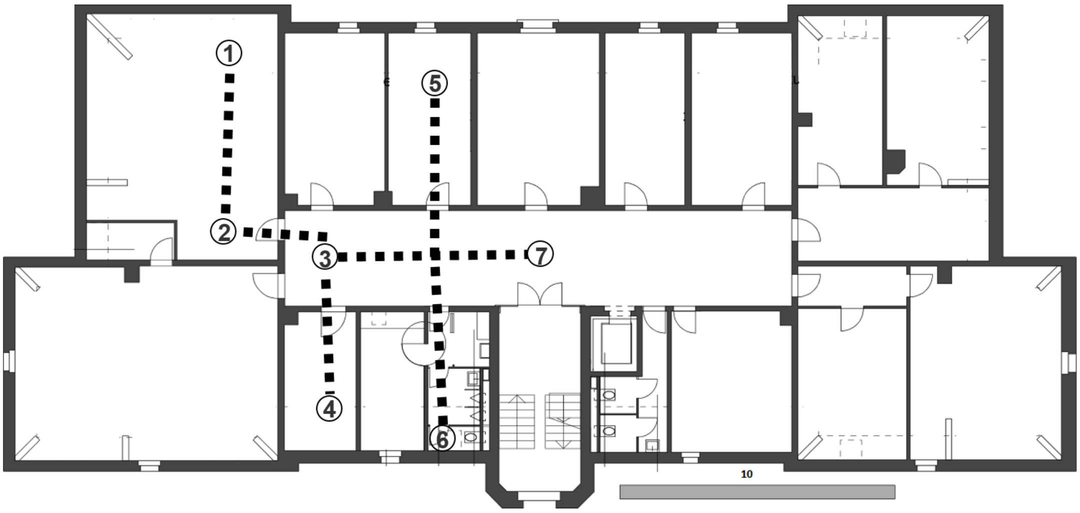

FIGURE A.2: Checkpoints and path of trajectory 2.

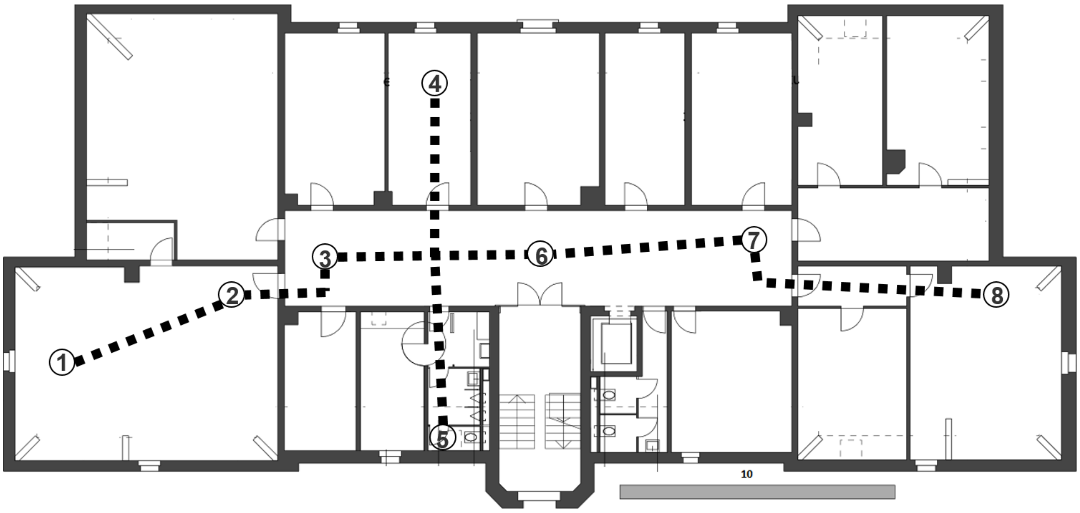
1
2
3
4
5
6
7
8
10

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

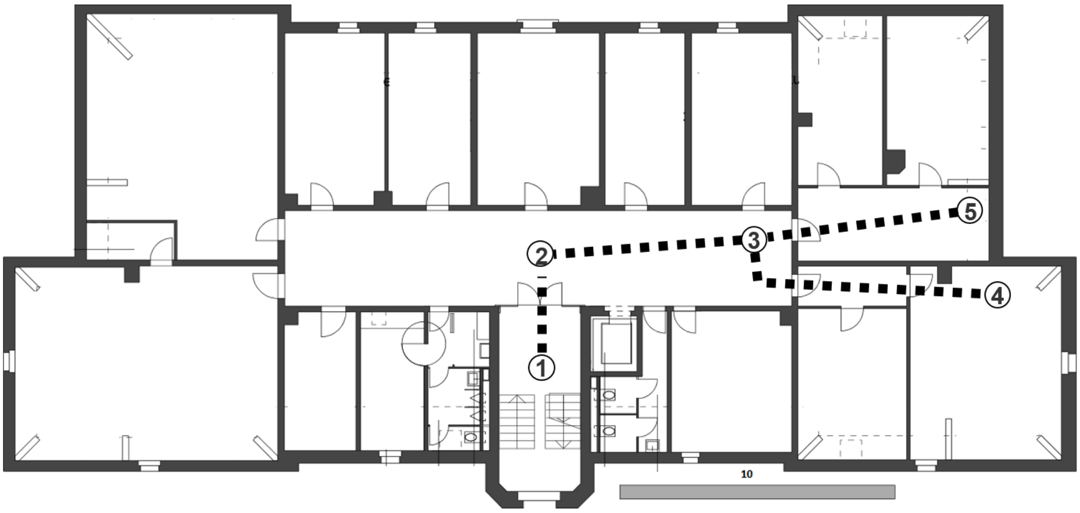

FIGURE A.4: Checkpoints and path of trajectory 4.

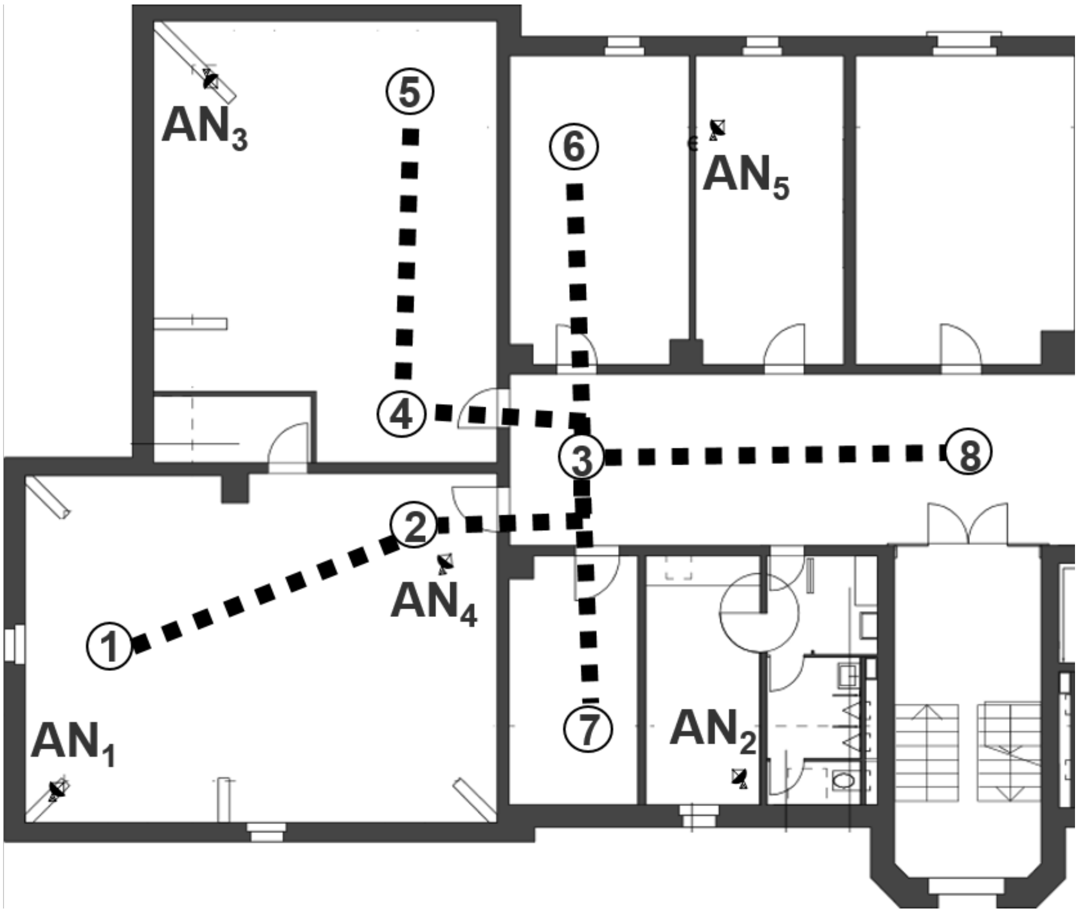
AN$_3$
⑤
⑥
AN$_5$
④
③
⑧
②
AN$_4$
①
AN$_1$
⑦
AN$_2$

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

Bibliography

[1] J.L. Carrera V. et al. “A Real-time Indoor Tracking System in Smartphones”. In: (2016).

[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: (2018).

[4] F. Dieter W. Burgard and S. Thrun. “Markov localization for mobile robots in dynamic environments”. In: *Artificial Inteligence Research* 11.1 (1999), pp. 391–427.

[5] G. Cleary and E.Trigg. “An instance-based learner using an entropic distance measure”. In: *12th International Conference on Machine Learning* (1995), pp. 108–114.

[6] *Computernetze Vorlesungsunterlagen Uebertragungsmedien*. HS: 2015.

[7] *D.WEKA Class multilayerperceptron*. `http://weka.sourceforge.net/doc.dev/`. Accessed: 2018-03-01.

[8] G. Forney. “The viterbi algorithm”. In: *Proceedings of the IEEE* 61 (1973), pp. 268–278.

[9] *Gartner Gartner Says Worldwide Sales of Smartphones Returned to Growth in First Quarter of 2018*. `https://www.gartner.com/newsroom/id/3876865`. Accessed: 2018-07-18.

[10] Ayhan Bozkurt Kerem Ozsoy and Ibrahim Tekin. “Indoor Positioning Based on Global Positioning System Signals”. In: 11.2 (2013).

[11] Richard Kerner. “The Thales experiment”. In: 11.3 (2017), pp. 1–2. URL: `https://arxiv.org/pdf/1712.06016.pdf`.

[12] *Raspberry Pi Symmetrical double-sided two-way ranging*. `https://www.raspberrypi.org/products/raspberry-pi-3-model-b/`. Accessed: 2018-07-31.

[13] R.Quinlan. “Programs for Machine Learning”. In: *San Mateo CA Morgan Kaufmann Publishers* (1993).

[14] *SEQUITUR InGPS Lite Beginners Guide and Application Examples*. SeqInGPSLite BeginnersGuide R1 2 pdf. No note.

[15] *SEQUITUR InGPS Lite User Manual and Application Programming Interface*. SeqInGPSLite UserManual R1 1 pdf. No note.

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

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

[18] *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.

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

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