Masterarbeit

Indoor Pedestrian Routing on Mobile Devices

Clemens Bauer

Institut für Technische Informatik
Technische Universität Graz
Vorstand: Univ.-Prof. Dipl.-Inform. Dr. sc. ETH Kay Uwe Römer

Forschungsgruppe für Fernerkundung und Geoinformation
Joanneum Research Digital
Leiter: Univ.-Prof. Dipl.-Forstw. Dr. Mathias Schardt

Begutachter: A.o.Univ.-Prof. Dipl.-Ing. Dr. techn. Eugen Brenner
Betreuer: DI(FH) Patrick Luley

Graz, im März 2014

Kurzfassung

In größeren öffentlich zugänglichen Gebäuden erfolgt die Orientierung sowie die Navigation hauptsächlich über Wegweiser beziehungsweise über Orientierungskarten. Ein klar erkennbarer Nachteil dieser Navigationsmethode ist, dass diese Karten oft nicht interaktiv und nur an gewissen Stellen im Gebäude vorhanden sind. Somit muss sich der Nutzer an Orientierungspunkte halten und auch selbständig herausfinden, wo sein Standpunkt gerade ist. Ein Wechsel des Stockwerkes kann bei dieser Navigationsmethode zu Orientierungsproblemen führen und erzwingt somit eine Neufindung der jeweiligen Position des Benutzers. In dieser Arbeit wurde ein Ansatz für die Positionierung mit Hilfe von Trägheitsnavigation und Radiofunkwellen verfolgt. Dazu wurden für das Android Betriebssystem eine Aufnahme-App sowie eine Positionierungs- und Routing-App entwickelt. Des Weiteren wurde ein Routingkonzept implementiert, welches die Wegefindung über mehrere Stockwerke hinweg ermöglicht. Die Routingdaten werden durch Metadaten ergänzt, um so eine genauere Wegbeschreibung wiedergeben zu können. Die bereits vorhandenen WiFi-Signale werden an selbstgewählten Positionen aufgezeichnet und in einem späteren Schritt wird daraus eine georeferenzierte Signalkarte erstellt. Durch diese Signalkarte kann der Client die aktuelle Position errechnen. Zur Verbesserung der Genauigkeit werden auerdem die Daten von Beschleunigungssensoren und Gyroskopen verwendet. Über einen Filter werden diese Daten miteinander abgeglichen. Im Weiteren wurde in dieser Arbeit der Ansatz verfolgt den Benutzer frei wählen zu lassen, welchen Weg er bis zum Ziel verfolgen möchte. Durch verschiedene Profile wird es dem Benutzer ermöglicht, seinen Weg individuell zu gestalten. Beispielsweise ist dadurch für Rollstuhlfahrer eine Wegeberechnung unter Ausschluss von Treppen möglich. Somit wurde eine Navigationsmethode entwickelt, welche einerseits klar auf die Bedürfnisse seiner unterschiedlichen Benutzer eingeht und andererseits den jeweils kürzesten Weg errechnen kann.

Keywords: Wegefindung in Gebäuden, Indoor Positionierung, WiFi, Android, Routing über mehrere Stockwerke, individuelles Routing, Vermeidung von geschlossenen Bereichen

Abstract

In large public accessible buildings orientation and navigation is mainly done by using signposts and orientation maps. A clearly recognisable disadvantage of this navigation method is that these maps are static and only available at distinctive points. Therefore the user has to continually figure out his current position. Furthermore he has to orient himself on points of reference. A simple change of a floor can ensure that the user has problems with orientation and force a repositioning.
In this thesis an approach for positioning of using inertial navigation and radio waves is used. Therefore, for the Android OS a recording app as well as positioning and routing app has been developed. Further, a routing concept has been implemented which allows routing over several floors. The routing data are extended by meta data used to provide a more precise description. The already existing WiFi signals are recorded at selected positions. In a further step, a georeferenced map is created by using this data. With this map the client determines the current position later. To improve accuracy the data from an accelerometer and a gyroscope are used. These data are fused with the WiFi data in a filter to get a more exact result.
Furthermore, this approach allows the possibility to give the user the opportunity to choose the route. Several profiles allow the user to get an individual route. For example an individual in a wheelchair can use a route without stairways. Additionally, a navigation method has been developed which calculates the shortest path with consideration of the users requirements.

Keywords: indoor routing, indoor positioning, WiFi, Android, multilevel routing, individual routing, avoid areas

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am …………………………… ………………………………………………..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

…………………………… ………………………………………………..
date (signature)

Acknowledgements

This thesis has been written at Institute for Technical Informatics at Graz, University of Technology in year 2014. I want to thank my supervisor Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Eugen Brenner who made this thesis possible.

This thesis has been supported by Joanneum Research Institute Digital at Remote Sensing and Geoinformation Group. This company gave me the opportunity to get a whole year to do research on this important topic. At this point I want to thank my advisors DI(FH) Patrick Luley and DI Stefan Ladstätter at Joanneum Research, who supported and helped me to finish this thesis. Without their help and input this thesis would not be finished yet. Furthermore, I want to thank DI Alexander Almer who shared his office with me and gave me a lot of tips and support. He was always keen to support me if there was a problem to handle with the administration. Additionally, he was a good discussion partner for topics concerning this thesis.

I also want to thank my family who followed the citation of Benjamin Franklin, who once said that an investment in knowledge pays the best interest. I want to say thank you, as you made it possible for me to receive my MSc degree. Furthermore, I want to say thank you to my grandparents who also supported me and helped me to finish my studies.

In addition I want to thank my friends who helped and supported me during my studies and the process of writing this master thesis. I also want to thank my friends around the globe who spent their spare time reading my thesis and correcting it.

Last but not least I thank my girlfriend Victoria. She gets the greatest thanks of all. She is the most loving and best person in my life. She supported me and understood this stressful phase of my life. What have I done without you? I thank you very much.

Graz, March 2014 Clemens Bauer

Contents

1 Introduction **1**
1.1 Problem Description and Motivation . . . 1
1.2 Structure of Thesis . . . 2
1.3 Related Work . . . 3
1.4 Summary . . . 6

2 Fundamentals **7**
2.1 Current Technical Solutions . . . 7
2.1.1 WiFi and Bluetooth . . . 7
2.1.2 Geomagnetism . . . 10
2.1.3 Acoustics . . . 12
2.2 Positioning Algorithms . . . 13
2.2.1 (Weighted) K-Nearest Neighbours . . . 13
2.2.2 Bayesian Algorithm . . . 14
2.2.3 All four directions Algorithm . . . 14
2.2.4 RBF-based Positioning Method . . . 14
2.2.5 Subtract on Negative Add on Positive (SNAP) Algorithm . . . 15
2.2.6 Method of Isolines . . . 16
2.3 Inertial Navigation . . . 16
2.4 Filter and Map matching . . . 17
2.4.1 Kalman Filter . . . 18
2.4.2 Particle Filter . . . 18
2.4.3 Map matching . . . 19
2.5 Indoor Multilevel Routing . . . 21
2.6 Individual Routing . . . 21
2.7 Subjective Sense of Security . . . 24
2.8 Summary . . . 26

3 Implementation for Mobile Devices **28**
3.1 Indoor Positioning . . . 29
3.2 Recording App . . . 30
3.3 Processing and Distribution Server . . . 40
3.4 Evaluation App . . . 42
3.4.1 Real-time Localisation Framework . . . 43
3.4.2 Multilevel Routing . . . 45
3.4.3 Individual Routing . . . 46
3.4.4 Subjective Sense of Security . . . 47
3.4.5 Guidance . . . 50
3.5 Summary . . . 51

4 Benchmark **53**
4.1 Test Area 53
4.2 Positioning 54
4.2.1 Only WiFi 54
4.2.2 WiFi - Four Direction Mode 62
4.2.3 Multisensor Fusion 65
4.3 Routing 68
4.3.1 Individual Routing 68
4.3.2 Subjective Sense of Security 69
4.4 Summary 70

5 Conclusion **72**
5.1 Summary 72
5.2 Future Work 73

Bibliography **74**

List of Figures

2.1 The left image shows a zone of possible positions. The right image shows an unambiguous point of intersection. 8
2.2 Figure is showing how acoustic reflection is working. 13
2.3 Image shows the architecture of the Radial Basis Function network. [LKP09] 15
2.4 Comparison of different algorithms used by Airplace [CL11]. 16
2.5 This figure show how the Strap down algorithm works. It gets the data from the motion and rotation sensors and determines the current position, velocity and attitude [Eck14]. 17
2.6 Overview about the Kalman filter [Aim14]. 18
2.7 Fingerprint Kalman filter algorithm [SALP09]. 19
2.8 Code of Particle filter, adapted to use with WiFi fingerprints [IK12]. 20
2.9 This image is showing how a particle filter is working [TFBD01]. 20
2.10 This image shows how the algorithm described in [SSO08] extracts a floor plan into a multilevel graph. 22

3.1 Overview about the JRindoor system. 28
3.2 Figure shows the workflow of the recording app. 30
3.3 The user has to select the folder where the images of the floors are stored. The name and address of the building can be chosen by the user. Furthermore these data can be also loaded from the configuration file, if they are provided in the configuration file. 32
3.4 The user can move the balloons to the edges of the building. The WGS84 coordinates of each balloon are stored in the configuration file. 32
3.5 In this step the user can chose a new origin point. The new point is shown in the floor plan. 34
3.6 The user can chose the number of samples which should be recorded at each position. Additionally a time interval between these recordings can be set. . 34
3.7 Different zones can be defined in the configuration. This is useful to assign a position to a zone. This is prior used for routing points. 35
3.8 The configuration of the sever connection is set in this step. 35
3.9 The user can chose between a normal recording and the four direction mode. This option is selected at the very first start of this step. 36
3.10 In this step the user records and manage the WiFi data. 36
3.15 Conversion from WGS84 coordinates into UTM. 36
3.11 In this step the user records the points for the routing graph. Besides, additional information can be added. 37
3.12 In this step the points are connected to get a routing graph. Besides metadata can be added to each point. 37

3.13 The connection of the points is done by selecting the node which should be connected. 38
3.14 Finally in this step the user can upload the data to the processing and distribution server. 38
3.16 Overview about the database for the recording app. 39
3.17 GUI of Server. 40
3.18 SQLite database output of the server processing. 41
3.19 Two possibilities are available. One is to download the building data from the server. This is done by using the provided information for the server. Second is to select existing data from the app. 43
3.20 The current building can be selected in this list. 43
3.21 This image shows the initial state of the app. The green arrow is the current position of the user. 44
3.22 This screenshot shows the menu of the app. 44
3.23 This figure shows the coordinate system of the Android OS. The used axis for the compass and the orientation of the user is the green x-axis. The used value is the azimuth value of the gyroscope. [Inc14b] 45
3.24 Connected nodes at a floor. 46
3.25 Figure shows the node connections between some levels. 46
3.26 Figure shows the overview about open/closed areas. 48
3.27 The screenshot shows the route to floor four under consideration of the usage of the elevator. 51
3.28 Screenshot shows the route to the destination. The guidance provides a textual interpretation of the way. 51
3.29 This chart shows the flow of the algorithm to generate the guiding instructions. 52

4.1 Test area in the fourth floor. 53
4.2 Test area “staircase” at each floor. 53
4.3 Illustrates the calculation of the position error. 54
4.4 Figure shows the deviation between the user set location and the real recording location. The cross in the centre of the circle is the position of the user. The measured radius around this location is about 30 cm. The red lines crossing the circle are the positions where the device is held. 56
4.5 Diagram shows the average positioning error of each algorithm for the first test. The result is split over each floor. 57
4.6 Diagram shows the adjusted average positioning error for the first test. Wrong detected floors have been excluded. 57
4.7 Diagram shows the average positioning error of each algorithm for the second test. The result is split over each floor. 57
4.8 Diagram shows the adjusted average positioning error for the second test. Wrong detected floors have been excluded. 57
4.9 Diagram shows the average positioning error of each algorithm for the third test. The result is split over each floor. 57
4.10 Diagram shows the adjusted average positioning error for the third test. Wrong detected floors have been excluded. 57
4.11 Diagram shows min and max values determined for each floor for KNN. . . 59

4.12 Digram shows min and max values determined for each floor for WKNN. . 59
4.13 Diagram shows how often the algorithm detected the wrong floor. 60
4.14 Error detection after recorded three times more samples at ground floor than at all other floors. 60
4.15 Floor detection error by using the same number of training samples at each floor. 60
4.16 Overview about the number of average detected APs in the elevator and in front of the elevator. 62
4.17 Diagram shows the evaluation of the direction of KNN algorithm. 63
4.18 Diagram shows the evaluation of the direction of WKNN algorithm. 63
4.19 Diagram shows the evaluation of the direction of MAP algorithm. 63
4.20 Diagram shows the evaluation of the direction of MMSE algorithm. 63
4.21 Diagram compares the positioning error by one and four direction evaluation mode. 65
4.22 Comparison of routing algorithms with test data of different directions. . . 65
4.23 Comparison of routing algorithms with test data of different directions. . . 65
4.24 Dead reckoning track in the test area. 66
4.25 Track computed by WiFi data using the one direction radiomap. 66
4.26 Track computed by WiFi data using the four direction radiomap. 66
4.27 Data fusion of MAP evaluation data by using one direction radiomap. . . . 67
4.28 Data fusion of MMSE evaluation data by using one direction radiomap. . . 67
4.29 Data fusion of MAP evaluation data by using four direction radiomap. . . . 67
4.30 Data fusion of MMSE evaluation data by using four direction radiomap. . . 67
4.31 In this figure the individual route using the stairs is shown. 69
4.32 In this figure the individual route using the elevator is shown. 69
4.33 The shown route in this figure is the shortest path from current position to the destination. In this route there exists no closed area. 70
4.34 The shown route in this figure avoids the closed door. Therefore the shortest path is to use the way through the next office. 70
4.35 Information about the closed area. 71
4.36 The user can activate this function by clicking an entry in the menu. If the option is activated, the route to avoid closed areas is shown. 71

List of Tables

3.1 The table gives an overview about the states of the four direction mode. . . 37
3.2 Table shows when the algorithm directs the user in another direction. . . . 50

4.1 Table gives an overview about the average positioning errors for each algorithm used in the first test. 58
4.2 Overview about recorded training and test samples at each floor. 58
4.3 Table gives an overview about the average positioning errors for each algorithm used in the second test. 58
4.4 Table shows the number of training and test data recorded for each floor. . 58
4.5 Table gives an overview about the average positioning errors for each algorithm used in the third test. 58
4.6 Table shows the number of training and test data recorded for each floor. . 58
4.7 Overview about the average RSS in each floor at Steyrergasse 17. 61

You can't always get what you want, but if you try sometimes, you might find, you get what you need.

(Mick Jagger)

1 Introduction

Smartphones became permanent colleagues in our life. Most people are using their smartphone not only for phoning and texting, but there are also a lot of apps available for various approaches. These apps are supported by built-in features in the device. A few of these features are different sensors. Such sensors can be used to illustrate the environment in apps. Altitude sensors can be used to determine the current altitude of the device. Accelerometers and gyroscopes can be used to determine the current position of a moving object. These sensors are used in vehicles or airplanes in combination with a navigation computer. The navigation computer is calculating the current position by using this data. There are many apps available which are using these sensors for navigation purposes. Further built-in features are communication systems (e.g. WiFi, Bluetooth). These features can be used to connect to other devices or to the Internet, but it can also be used to determine the position of the device.
Navigation and location based services became an often used part in smartphones. To find the shortest path to the destination a smartphone is utilised by the user. Such apps or services can calculate routes by car, bicycle, public transport, or just by foot. Indoor routing and positioning services are not so popular. These services or apps follow specific approaches. For each building the data has to be provided individually. Besides, there are different technologies available which are not compatible with all smartphones.
In 2011 [Meu11] Google introduced indoor navigation for some buildings and made indoor navigation available for the crowd. Several public buildings are available in Google Maps. The current restriction is that it is a closed source system and cannot be extended. While outdoor navigation is already solved, indoor navigation is not. In Section 1.3 various related solutions for indoor positioning and navigation are described and discussed. Therefore, this thesis shows some approaches to improve indoor navigation and routing.

1.1 Problem Description and Motivation

Navigation in buildings is not always easy. The navigation is mostly done by signposts or "You are here maps" at some distinctive points. Such navigation aids are not helpful if the person is anywhere in the building without a sign and does not know the right way. Outside it is generally easy to find the desired way by using a smartphone. If GPS or the mobile phone network is available, the current position can be determined. However, in buildings it is tricky to determine the current position by using these technologies. GPS for indoor positioning in most cases is not available. In cities the location of a mobile phone can be accurately determined within 100 to 500 m [Sch13]. This is too vague for indoor positioning. Therefore, other technologies have to be investigated to get an indoor position precise enough to obtain 3-5 m. To solve this problem a localisation and positioning framework has to be developed. This should use existing radio signals to determine the position.

For a more precise result a filter will be needed. The calculated position may not be on the trajectory of the pedestrian. Therefore, map matching can calculate an approximate position of the user. This is done with consideration of two inputs. In this case one input is the WiFi data and the other should be taken from the output of the navigation computer. This process is done in the filter. The motivation is to develop a mobile application for smartphones, that is able to determine the current position based on existing radio signals. The system should work independent from the Internet. Therefore, the necessary data has to be downloaded from a server before starting indoor positioning. Only localisation in buildings does not help the user too much. The user needs to be guided to their destination in the building. This guidance should include finding the right floor and the right room or office. The idea is to include a routing system with guidance. This routing system should be able to find the way in a multi-floor building. Buildings have stairs and elevators also. People who are not able to use the stairs should be guided to use the elevator. Therefore, two different routing profiles have to be included to guide the user to their destination: One profile for stairs and another for elevator usage. An additional aspect considered in routing is the avoidance of particular areas. For example, the building has two stairways and one is closed due to construction work from 2 pm until midnight. With proper guidance, the user will be guided around and will not see this construction site if they have to use this stairway between 2 pm and midnight. The motivation is to include a routing system which is able to use the mentioned points to guide the user through the building under consideration to show him the shortest path. Furthermore, the reason for the closed area should be displayed because the user should to be informed, why they cannot use this particular area.

1.2 Structure of Thesis

This thesis starts in Chapter 2 with an overview about the state of the art techniques for positioning. In the first section current technical solutions and different methods for positioning are discussed. These solutions are based on WiFi, Bluetooth, Geomagnetism, and Acoustics. The most used algorithms to determine the positions are explained. Inertial navigation, a system for infrastructure-less positioning is shown.
Also, solutions for filtering and map matching are described. In the later sections different approaches for routing are discussed. In the last section the possibility of subjective sense of security by walking is discussed. This topic can help to increase the sense of security of persons outside and in buildings.
The practical implementation of this work is shown and explained in Chapter 3. The app system consists of two apps and a platform independent local server. One app is used to record all data for positioning and routing. The second app is used to evaluate the current position. Furthermore, it helps the user to find the way to their goal. The server is used to process the data and to distribute it to the positioning app. In the sections, the implementation of routing related algorithms and context aware controls are shown. Additionally, the post processing localisation framework is explained. This module is used to do positioning evaluation after positioning data have been recorded.
Chapter 4 discusses and shows the results of the thesis. It shows the accuracy of floor detection and the average positioning error. Further it evaluates the four direction algorithm

and their average positioning error. It shows improvements reached by filter implementation and by using inertial navigation. Furthermore it shows the advantages of individual routing and the possibility of excluding insecure and closed areas in buildings. A comparison of routing solutions that are not using such technologies is done.
The last chapter which is Chapter 5 concludes the thesis and gives an outlook about possible improvements and extensions.

1.3 Related Work

This section gives an overview of different available indoor positioning and routing systems on smartphones. Most applications for indoor positioning are based on Wireless LAN (WLAN) radio signal determination.

Indoor Pedestrian Navigation System
A new approach of indoor navigation is introduced in [SCM10]. The floor plan is provided online and the user can download the floor plan via a datamatrix from the web. The data read from the phones sensors, combined with the reference map of the place, gives the actual position of the user without connecting to any external or pre-installed positioning system such as GPS, RFID, or WiFi trilateration using the dead reckoning technique. The prototype of this system, uses the data from the motion sensors embedded in the smartphone to compute the correct position of the user based on a known initial location. The way of the user is drawn step by step in the downloaded map. [SCM10]

Footpath
In [PFS+12] two important parts of indoor navigation are shown. First, it has a focus on positioning and orientation. Second, it has a focus on using maps for smartphone navigation. In the second part the focus is on the general ability of indoor maps. The problem is that most buildings do not provide exact maps. This can have several reasons (e.g. security reasons). The authors of this paper suggest the possibility how to receive the data from the building.
After obtaining map material either from a public source or via a photographed evacuation plan, it presents the user a selection of destinations, to which they can navigate. The smartphone then detects the steps of the user using the inbuilt sensors. If this is not possible, the rear-facing camera is used instead to measure the optical row and derive pseudo-steps. On the basis of this speed and heading detection, the resulting steps are plotted onto the available map. The paper shows that this system is accurate enough to provide the user with accurate turn by turn navigation instructions. [PFS+12]

Footpath - Wheelchair extension
A new challenge is to guide wheel chair drivers. This is described in [BLGSW12]. Two different position systems are used. Positioning will be carried out by WiFi using fingerprints or by radio independent systems based on motion and direction sensors. The system can detect three classes of speed: stop, slow and fast.

Microsoft RADAR
In 1999 Microsoft Research introduced the first WiFi based positioning system called RADAR. RADAR is a radio frequency (RF) based system for locating and tracking users inside buildings. It uses signal strength information gathered at multiple receiver locations to triangulate the coordinates of the user. Triangulation is done by using both empirically-determined and theoretically computed signal strength information. The used method in this paper is to record information about the radio signal as a function of the location of the user. The signal information is used to construct and validate models for signal propagation during offline analysis as well as to infer the location of a user in real-time. It is a system for locating and tracking users inside a building. RADAR is based on empirical signal strength measurements as well as a simple yet effective signal propagation model. While the empirical method is superior in terms of accuracy, the signal propagation method makes deployment easier. The despite the hostile nature of the radio channel has been shown in this paper. Further it shows that with this system it is possible to locate and track users with a high degree of accuracy. The median resolution of the RADAR system is in the range of 2 to 3 meters, about the size of a typical office room. The results indicate that it is possible to build an interesting class of location-aware services, such as printing to the nearest printer, navigating through a building, etc., on an RF wireless LAN, thereby adding value to such a network. [BP00]

Airplace
Airplace is an indoor positioning application for Android smartphones. It consists of the RSS Logger app, a distribution server and a tracker app. The RSS Logger is an application developed around the Android RSS API for scanning and recording data samples in specific locations at predefined intervals. These samples contain the MAC addresses and RSS levels (in dBm) of all neighbouring WLAN Access Points (AP), as well as the coordinates of the location where the user initiated the recording. In this paper they have implemented several fingerprint-based algorithms, including the deterministic K-Nearest Neighbour (KNN) and Weighted K-Nearest Neighbour (WKNN) algorithms, as well as the probabilistic Maximum A Posteriori (MAP) and Minimum Mean Square Error (MMSE) algorithms. A distribution server is introduced to calculate the radiomap and the parameters. [LCC+12]

IndoorAtlas
In [Hav12] an indoor positioning system based on geomagentic fields is presented. In IndoorAtlas' location technology, anomalies (fluctuations) of ambient magnetic fields are utilised in indoor positioning. This has been facilitated by modern smartphones and the rapid development of sensor technology. In this report they introduce a complete software solution:

- Add and manage floor plans with IndoorAtlas Floor Plans™web application
- Collect magnetic field data with IndoorAtlas Map Creator™mobile application
- Use IndoorAtlas API to use the location service

IndoorAtlas is a cloud based indoor positioning service. The Android app is used to record the magnetic data and to generate the geomagnetic indoor map. The company provides only an API to use the software. [Hav12]

Indoo.rs
The indoor positioning system from indoo.rs [ind14] is smartphone based. It uses WiFi and Bluetooth to determine the current position of the mobile device. Indoo.rs is providing a cloud solution mainly for companies to use the framework provided for Android and iOS. The system works with adding the floor plan to the provided software on the laptop. Furthermore all areas and specific details for the floor are added in the software. Specific details show where are the windows and the width of walls. After this is done, the signals can be recorded by using the laptop. The data are stored in the cloud and can be used with the provided SDK (Software development kit). This SDK allows the user to create their own application. The disadvantages of this solution are the usage of a laptop, the cloud solution and necessity to provide full details of the building. The software is not easy to use. Therefore these details are difficult to add.

RedPIN
RedPIN from Federal Swiss Institute of Technology is a very early indoor positioning application developed for Android, iOS and Symbian. The researcher introduce a new concept called "asynchronous interval labeling" that addresses these problems in the context of user-generated place labels. By using an accelerometer to detect whether a device is moving or stationary, the system can continuously and unobtrusively learn from all radio measurements during a stationary period, thus greatly increasing the number of available samples. Movement information also allows the system to improve the user experience by deferring labeling to a later, more suitable moment. [Bol08, PB09]

RoughMaps
The components of the RoughMaps platform can be grouped into three main parts: the web application running on the server; the web administration interface also running on the server; and the client application that makes use of the RoughMaps client-API, running on a mobile device such as an Android smartphone. This research looks at a new mechanism for managing arbitrary symbolic maps and for providing this information in a manner of contextual value to the user. The platform is expected to also become an essential tool for the continued study of symbolic maps and their relevance and importance in personalised and context-aware mobile applications. [RW12]

SMARTPOS
SMARTPOS is an indoor positioning system based on deterministic 802.11 fingerprinting and a digital compass. The system considers the users orientation to avoid errors caused by the blocking effect of the human body. For location estimation it only takes into account the part of the fingerprint database that corresponds to the current orientation of the user. SMARTPOS achieves a mean position error of 1.16 m and a maximum position error of 2.74 m in a 250 square meter environment. [KW11]

1.4 Summary

This chapter gave an introduction to the topic. It shows that indoor positioning and routing is an important topic. It can help to make navigation in buildings more comfortable for the user.
The outcome of Section 1.1 is that technologies such as GPS and mobile phone networks are not as exact as WiFi in buildings could be. GPS has the disadvantage that it is not always available indoors. Furthermore, the calculated position is may not on the trajectory and has to be filtered to get a more accurate result. Three main problems about routing are discussed. These problems are multifloor routing, wayfinding using different profiles and the possibility to exclude areas.
In Section 1.3, different existing solutions for indoor positioning are shown. Some solutions for example RADAR and RedPin are outdated and currently there is no further development. The Roughmaps project had been shut down and the sources were no longer available on the Internet. Smartpos provided an accurate positioning, but the sources were also no longer available on the Internet. Others, for instance Indoo.rs and IndoorAtlas, are closed source and cannot be extended. The available APIs can include the systems in apps, but the data is processed by the closed source system. Footpath for example is an infrastructureless system which does not use WiFi. Because of this fact it cannot be used in this thesis. The goal of this thesis is to use an existing solution that supports WiFi positioning. Furthermore, it should be extendable by routing and guidance. In comparison with the other solutions shown, Airplace seems to be the most suitable according to the criteria presented. It is open source and can be extended further. A drawback of Airplace is that there is no further development and it does not fulfil the requirements. An advantage of Airplace is that it is based on WiFi positioning based on the fingerprinting method and the available version of Airplace contains four different positioning algorithms. To conclude, Airplace is most suitable to be improved and extended with the required features.

One cannot see into the future, but one can lay the foundation for things to come for the future is something one can build.

(Antoine de Saint-Exupèry)

2 Fundamentals

This chapter shows the different available solutions and approaches for indoor navigation. In these sections the state of the art technologies are represented. This chapter is organised as follows: In Section 2.1 several technologies which can be used for positioning are shown. It discusses the advantages and disadvantages of the different technologies. In Section 2.2 several positioning algorithms are discussed. Sensor fusion with inertial navigation is explained and discussed in Section 2.3. It is combined with an in Section 2.1 shown method to determine a precise location. Section 2.4 illustrates filter and map matching technologies along with different approaches and the usage of filters. In the last three sections 2.5,2.6 and 2.7 various routing options are discussed with details of different routing solutions.

2.1 Current Technical Solutions

This section discusses the current technical solutions. Technologies are examined which can be used for indoor positioning and localisation. All reviewed technologies are based on radio waves and magnetic fields. Only common and upcoming technologies, which do not need any additional infrastructure are discussed. WiFi hotspots are available in most buildings, therefore this kind of technology can be seen as common.

2.1.1 WiFi and Bluetooth

WiFi and Bluetooth are similar. Therefore positioning methods can be used for both. Both are using electromagnetic waves as transportation systems and are wireless. Bluetooth communicates using radio waves with frequencies between 2.402 GHz and 2.480 GHz, which is within the 2.4 GHz ISM frequency band, a frequency band that has been set aside for industrial, scientific and medical devices by international agreement. Bluetooth is divided into three classes, each of which has a different range [Bek12]. WiFi uses hotspots, while Bluetooth uses beacons.

Neighbourhood Recognition

The easiest and fastest possibility to determine the position is to detect if a reference point is within the receiving radius of an access point. If this is true, then the client will receive the coordinates of the reference point. In the case of several access points the lowest distance is determined by using the signal strength. This technique shows a mathematical error in the range of the radio communication. The conference paper [Jan07] shows that the mean of the positioning error is in half of the range. The advantage of this method is that it is easy to implement. The disadvantage is the large positioning error.

Multilateration

Trilateration is a technique which needs three senders to determine the position. If there are more than three senders, then it is called multilateration. In [BC09] a method for wireless LAN multilateration is introduced. This method finds the current position by overlapping the signal strength of the APs. The usage of the relation between signal strength and distance make it possible to estimate the distance between the transmitter and the device. The Formula 2.1 shows the relation between signal strength in dBm (P_t), the received signal (P_r), the wavelength λ, the path loss exponent n ($n = 2$ in free space) and the distance between sender and receiver d.

$$P_r = P_t + 20log\frac{\lambda}{4\pi} + 10nlog\frac{1}{d} \tag{2.1}$$

The distance is determined by creating circles around the sender. The device for which the position should be estimated is where the circles intersect. The precision depends if the intersection point is unambiguous. The best information is given, if the circuits are intersecting in only one point (good information). If this case is not given, a greater zone with position possibilities are available (bad information). Both cases can be seen in Figure 2.1. In [Bek12] the described method is used for Bluetooth positioning. [BC09] points out that the number of connected devices and changes in the environment have influence on the precision of the position result. A big problem is the reflection of RF signals. This paper shows the processing steps how to filter disturbed RF signals to make positioning possible.

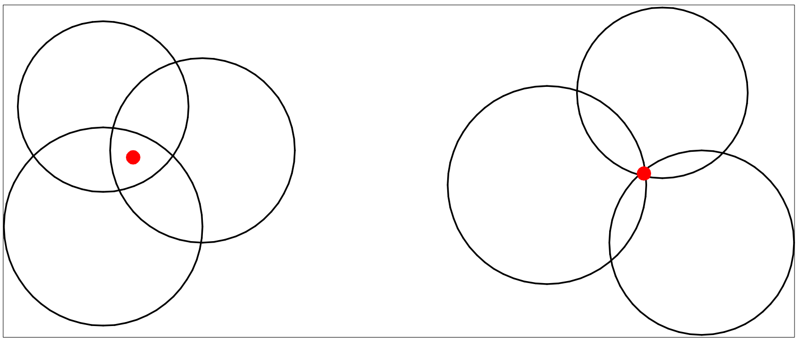

Figure 2.1: The left image shows a zone of possible positions. The right image shows an unambiguous point of intersection.

Distance Measuring using the Signal Strength and Signal Duration

To determine a two dimensional position it is necessary to know at least three points of references. In general a measuring error at the determination of the distance is very critical. This is because the error can be added by the calculation of the position and under circumstances an inconvenient result can be determined. [Jan07]
With mobile devices with included Bluetooth and WLAN the signal attenuation of the signal level can be read directly from the transmitter as RSSI value. The measured value can somewhat vary depending on the type of device. Mathematically, the signal strength

is decreasing in squared quantity to the distance. The distance to the receiver can be determined by measuring the exact time between the beginning of the transmission at the sender and the moment of receiving the first data at the receiver. This method is called Time of Arrival (ToA). The usage of this method requires a precise timewise synchronisation of all involved devices. A short deviation with the synchronisation can lead to a false positioning result. This method is difficult to implement because of the synchronisation. [Jan07]

radiomap

A radiomap is a database which contains for each recorded position all available MAC addresses with the corresponding signal strength RSS(Received Signal Strength). These data are recorded in the area where the user should be positioned. [LJYH13]
The disadvantage of this method is that it is necessary to keep the database up to date. Changes in the WiFi network by adding or removing APs can cause positioning problems. A complete new recording of the radiomap is very expensive and time intensive, especially in big buildings (e.g. multifloor building). In [LJYH13] a new method is presented to keep radiomaps up to date, even by very few changes. The method is using a scoring system. This system is based on user feedback. If a new AP has been found, the score will be increased by one. If several users found this AP, the score will be raised further. After the threshold of a score value is reached, the AP is added to the radiomap. It also works in reverse order to remove AP from the radiomap.

Fingerprinting

To avoid measurement errors, the usage of radio fingerprints improves the positioning process. A radiomap is recorded in the first recording phase in the area where the user will be positioned. This recording is done by using a plan of the building. The recorded data are stored in a radio map. For each recording point the MAC address and the signal strength are stored. Variations in the electromagnetical signal strength can influence the positioning precision. To avoid such variations two recording possibilities are available: The deterministic method uses mean values of the signal strength to mitigate deviations. The probabilistic approach captures deviations as probability distribution. The storage of all data is mostly done with a database and finishes the training phase. The determination of positions is done in live operation by comparing the current recorded data with the stored data in the database. [Jan07]

Client based vs. Server based

The position determination can be done at the Mobile Device (client) or at the Access Point (AP). A client server based method uses the received signal strength of the infrastructure to determine the position. The infrastructure are the APs in the building. The client receives regularly signals in the form of data transfers or beacons. The AP has to send out this signals regularly. In case of WiFi networks this is guaranteed by the IEEE 802.11 standard. In this standard it is defined that beacons have to be sent out regularly. The client calculates the position by the determined k values, where k is the number of the reachable AP.

It is called server based positioning when the position determination is done by the infrastructure. In this case the clients send packages in defined intervals. The APs are configured to read the signal strength from the received packages. With this information, the infrastructure is able to calculate the position of the device. In contrast to the client method, at where this information is available at a central point, the signal strength has to be sent to the AP. The APs use this information to calculate the current position. [Jan07]

Cell based technique

Bluetooth provides the possibility to determine the position by cells. [Bek12] The approach shown in [Cha09] is based on the visibility of Bluetooth beacons to determine the position of the mobile device. This method divides the building into different sections, depending on the range of the beacons. The position can be determined by which beacon is able to see the mobile device and which is not.

2.1.2 Geomagnetism

A magnetometer can be used to detect the location of geomagnetism indoors. Such a sensor is built-in to most smartphones today. The magnetometer is widely used as a part of the inertial measuring unit (IMU) as a source of heading correction. [KPR10] About 2000 years ago the Chinese invented the compass. The compass uses the magnetic field to show the directions [Wen89]. The arrow of the compass points alway to north and therefore the compass is able to show the directions. The idea using such a sensor is therefore borrowed from both, the usage of a compass and the world of animals. Researchers have focused on the fact that some animals use Earth's magnetic field for navigation and orientation. A few animals, such as spiny lobsters and green sea turtle, are not only able to detect the direction of Earth's magnetic field, but also able to sense their true position relative to their destination, which means these particular animals are able to derive positional information from local cues that arise from the local anomalies of Earth's magnetic field. It is a fact that steels and reinforced concrete structures of buildings cause distortion of the geomagnetic field and create anomalies. Magnetic field variations inside buildings are found in iron, cobalt, or nickel and also occur from man-made sources such as steel structures, electric power systems, and electronic appliances. If these variations or anomalies are identified, they provide a unique fingerprint or profile for places inside buildings where they exist. [GSDM11]
In general, magnetic distortions have been considered as undesired property for detecting the heading using an electric compass. [KKYK12] Therefore it can be used for localisation in buildings. Anomalies can be recorded as fingerprints in a map. Hence, the more variable the local anomalies, the more unique the fingerprint and the better the positioning. [KPR10]
A magnetometer receives three coordinates (x,y,z) and the field strength measured in μT. Each one of the measured coordinate represents the value of the magnetic field along one direction of the phones frame ($x_{phone}, y_{phone}, z_{phone}$) which redirects to $P_B^{measured}$. [KKYK12] Phone applications do not usually have access to the raw output of the magnetic sensor. Instead, they receive values resulting from a filtering, often performed on the sensor chip itself. [LGT12]
A map with the magnetic field of the room will be required for positioning. The easiest

method is the 2D mapping method. In [LGT12] a 2D map has been manually set up by covering the room in serpentines twice. The paper [LGDR12] shows that the accuracy is good even if there are only two elements recorded. The accuracy in small grids is quite better than in large grids. It also depends on the number of recorded elements. The medium accuracy is between 40 cm and 2 m, depending on the grid size.
The paper [KPR10] presents a long time test of magnetic field changes. For this test three different rooms are used and a smartphone has been positioned on the floor to record the magnetic field for 24 hours. The results show that the field is constant. But after three months the test has been repeated, the results are differed from the previous results. All of the intensity values were smaller than those of the previous tests. However, the inclination values were very similar. Since the sensor was used by other researchers (including overseas), a reasonable explanation is that the configuration of the sensor (the value to normalise the intensities) was changed. [KPR10]
In [LGDR12] the usage of magnetic density for fingerprint positioning is discussed. The results of the tests show that the indoor geomagnetic field is stable, and that characteristics change with location. The usage of the fingerprinting method requires that the device has to know true north or the magnetic north. The sensor needs two phases to provide results - the training phase and the positioning phase. The results of both should be the same. Furthermore the direction of gravity has to be known. That can be detected by the accelerometer and is important because it depends how the user is holding the smartphone. The relationship between the sensor-fixed coordinate system, the earth-fixed coordinate system and true north (X') are unknown. It is impossible to calculate the X' and Y' components of the field vector b. The Z' component can be calculated and the component in the X'Y' plane is the same no matter the direction of true north (X' should point to true north, but this direction is unknown). The magnitude of vector b can be easily obtained and θ can be calculated using the following Formula 2.2.

$$cos\left(\frac{\pi}{2} - \theta\right) = \frac{a_x \cdot b_x + a_y \cdot b_y + a_z \cdot b_z}{\sqrt{a_x^2 + a_x^2 + a_z^2}\sqrt{b_x^2 + b_y^2 + b_z^2}} \tag{2.2}$$

To apply a fingerprinting method, a database must be created in a so-called "training" phase. The data collected during tests described in the previous sections were used for training purposes. The small grid test was first investigated. Two databases were created, using the three elements (intensities in X, Y and Z directions) approach and the two elements (intensities in Z and XY plane) approach. Then one of the reference points was removed from the database and used as a test point. The basic nearest neighbour (NN) algorithm (using Manhattan distance or Euclidean distance) was used to estimate the position of the test point. [LGDR12]
Advantages of the fingerprint method are: no pre-deployed infrastructure is required, the magnetic field is everywhere and relatively stable, and change with the location.
The paper [LGDR12] highlighted on the one hand the advantages of this technology and some investigative tests. On the other hand it shows that if the correct building was unknown, using the geomagnetic field fingerprint alone could locate the user in a wrong building. This could happen if fingerprints are similar. Other information should therefore be utilised to constrain the area where the test point is located, or some other positioning

technology, such as WiFi, could be used in combination. The number of elements that can be used to create the fingerprint database are small, maximum three, though the number could fall to two in many applications. This makes the fingerprinting methodology not very reliable. The change of magnetic field with location is quite significant. Methods are needed to create the database in an efficient manner. Lastly, magnetic interferences should be considered, especially in areas close to elevators. The height of the sensor in the training phase and positioning phase could be completely different.
In [LGT12] a real time indoor localisation method that utilises a single 3-axis magnetometer to estimate the location of a handheld device is presented. Using an online particle filter, a localisation accuracy of 0.7 m in position and 25 degrees in orientation for the simple straight line trajectory could be achieved. The main limitation of this indoor localisation approach is the necessity of creating magnetic field maps. Therefore, a fast 2D mapping technique has been presented and demonstrated in a square room. Robots use geomagnetic anomalies to navigate through floors. SLAM (Simultaneous Localisation And Mapping) algorithm generates a map while a robot travels through an unknown space, and determines the current location simultaneously. [ZM11]

2.1.3 Acoustics

Sound waves can be used to determine the position. One of the most popular application is the usage of sonar on submarines. This technology is taken from nature. Bats and underwater mammals use acoustic waves to locate themselves. In the seminar thesis [Man08] an acoustic positioning system is introduced. Figure 2.2 shows how the measurement of signal propagation time works. The sender and receiver are the same device. In most applications ultrasonic is used. Ultrasonic cannot perceived by humans. A disadvantage of using this method is that velocity of acoustic waves depends on air flow, temperature and air humidity. This produces an error of about 1 mm per meter per degree of temperature. The second disadvantage is the low sampling rate. This sampling rate arises through slow spreading of acoustic waves.
In [RSA$^+$13] an acoustic positioning system in rooms is shown. The used technique is based on active sound fingerprinting and needs no infrastructure. Rooms and within-rooms positions are characterised by impulse response measurements. Using acoustic features of the impulse response and pattern classification, an estimation of the position is performed. The RoomSense App enables identifying an indoor localisation within one second. To evaluate the recognition performance of the RoomSense system the impulse response dataset is used. For the position classification, a Support Vector Machine (SVM) classifier with a Gaussian kernel is used. In a training phase, the training set feature vectors including the position and room labels were derived and SVMTrain was used to create pattern models for all rooms and within-rooms positions. In the testing phase SVMClassify used the stored models to classify a new feature vector FSEL regarding room and within-room position. The evaluation study in this paper has shown that RoomSense achieves a room-level accuracy of $> 98\%$ and a within-rooms positions accuracy of $> 96\%$.
Another method called ASSIST (Acoustic Self-calibrating System for Indoor Smartphone Tracking) is shown in [HZH$^+$12]. While the previous method is infrastructure independent, this system needs a WiFi network. The smartphone user is localised with small effort, affordable equipment and with high accuracy in indoor areas. The system uses

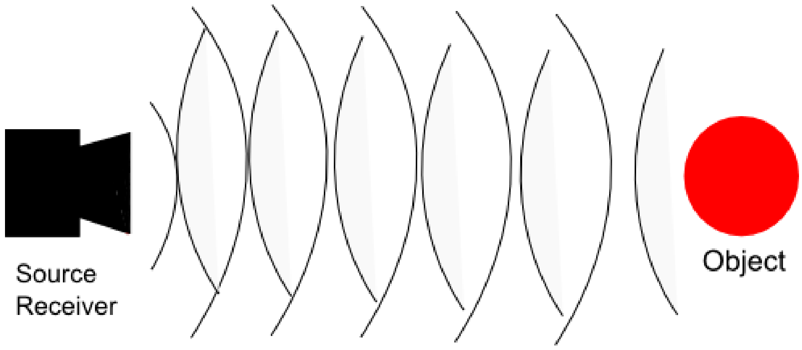

Figure 2.2: Figure is showing how acoustic reflection is working.

commercially available smartphones generating high pitched acoustic chirp signals beyond the audible range. The chirp signals are received by sound receivers which identify the specific sound produced from each smartphone. The receivers are connected to a WiFi network, to synchronise their clocks and exchange the time differences of arrival (TDoA) of the received chirps. In this way, using an iterative multilateration algorithm, the location of the smartphones can be calculated and the receiver positions are calibrated automatically. The sound signals are generated by an Android app. The user is assigned specific parameters, such that several devices can be distinguished by the appearance of the chirps. The position of the user is displayed on the smartphone in context of the environment, with a map and surrounding items. In [IDV13] researchers from EPFL Lausanne introduce an algorithm to reconstruct the 3D geometry of a convex polyhedral room from a few acoustic measurements. In this method they use the first-order echoes collected by a few microphones succeed to describe the dimensions of a room uniquely. The algorithm opens the possibility to make indoor positioning using a single source and a few microphones to detect the current location. In the experiments they used four microphones to detect the room. The paper shows that it is possible to describe a lecture hall with this method.

2.2 Positioning Algorithms

This section shows the most used positioning algorithms. Each algorithm uses different methods to determine the position of the device.

2.2.1 (Weighted) K-Nearest Neighbours

The (weighted) k-Nearest Neighbour algorithm uses the principle of a radiomap, which has to be recorded in advanced. In the radiomap or any other source stored signal data will be compared with the current recorded data. Using Formula 2.3 the Euclidean distance between the stored and the current recorded signals are calculated. RSS_i in the formula are the signals in the radiomap. S_i are the current recorded signals. In opposite to the unweighted k-Nearest Neighbour algorithm takes the weighted k positions instead of one. With the calculated results the algorithm estimates the current position. This is done by storing the calculated distances. The mean distance is calculated by using the calculated distance. [Zho05, YZP13]

$$d_i = \sqrt{\sum_{i=1}^{m} (RSS_i - S_i)^2} \tag{2.3}$$

2.2.2 Bayesian Algorithm

The maximum a posterior (MAP) and the minimum mean square error (MMSE) are algorithms which are based on the Bayesian approach. The algorithm calculates the maximum a posteriori probability with Formula 2.4.

$$posterior = \frac{prior * likelihood}{evidence} \tag{2.4}$$

$$p(l_i|s) = \frac{p(s|l_i)p(l_i)}{p(s)} = \frac{p(s|l_i)p(l_i)}{\sum_{i=1}^{i} p(s|l_i)p(l_i)} \tag{2.5}$$

$$P(s|l_i) = \prod_{j=1} p(s_j|l_i) \tag{2.6}$$

The Formulae 2.4 and 2.5 describe the Bayesian approach. This is calculated over an a-priori information with the likelihood function as the estimation value. The position l_i denotes the random variable. The corresponding probabilities are a-priori probabilities. The posteriori probabilities are estimated with these formulae. The estimated value is computed with the product for all positions calculated probabilities.
The MMSE algorithm stores the weight of the results and evaluate it. With this procedure weighted positions are calculated with which the probability can be evaluated. [Bek12, P.07]

2.2.3 All four directions Algorithm

In [Zho05] an algorithm is introduced which is able to detect the direction of the device. The algorithm is made for WiFi. The principle is to save WiFi data in the radiomap for each direction. For each recording the mean value is calculated and stored. In the evaluation step the Euclidean distance for each direction is calculated. The one with the shortest way is the current direction and position.

2.2.4 RBF-based Positioning Method

The presented approach in [LKP09] is based on a special class of Artifical Neuronal Networks (ANN), called Radial Basis Function (RBF) networks. They utilise location fingerprints that contain RSS measurements from several APs available in WLAN. The contribution of this paper is the presentation of an efficient localisation method based on RBF networks. They apply a clustering scheme to reduce the network size and computational overhead during the localisation process. Experimental results indicate that the proposed method provides more accurate location estimates compared to other approaches. Moreover, the underlying RBF architecture is scalable and can be easily applied to different WLAN setups, in which variable number of APs, reference points or fingerprints may be

available. In general, RBF networks are ANNs that have an input layer, a single hidden layer with non-linear radial, i.e. distance based, basis functions and an output layer. The architecture of a fully connected RBF network is depicted in Figure 2.3. In the context of indoor localisation using fingerprints, the RBF has n inputs, corresponding to the RSS measurements from all n available APs and two outputs representing the 2-dimensional coordinates. Experimental results indicate that adequate level of accuracy is achieved even with limited number of reference points, collected fingerprints or installed APs. The proposed method is scalable and can be easily applied to other environments. The heuristic used to set the value of β ensures that under different conditions the cRBF design interpolates the reference data smoothly and performs well during localisation. [LKP09]
Figure 2.4 shows the positioning accuracy of this algorithm in Airplace. It is a few cm more bad than all the MMSE algorithm.

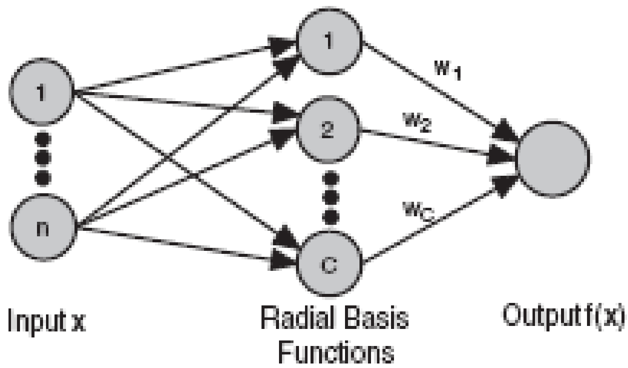

Figure 2.3: Image shows the architecture of the Radial Basis Function network. [LKP09]

2.2.5 Subtract on Negative Add on Positive (SNAP) Algorithm

In [LMP11] it is proposed to adapt the SNAP algorithm to the WLAN setup using only information of whether an AP is detected during positioning or not. Secondly, it shows how the accuracy of SNAP can be improved by introducing the idea of zones to exploit different RSS levels. This algorithm achieves a level of accuracy that is comparable to other well-known positioning methods, but is considerably simpler and much faster which is desirable for low power mobile devices in order to save valuable energy. The SNAP algorithm employs three main components to derive the unknown user location: Determination of a Region of Coverage (RoC), Likelihood Matrix and Location Estimation by calculating the maximum of the likelihood. The SNAP is improved to ignore the negative contributions of the failed APs. The outcome of the paper shows that the SNAPft-z has a low positioning error, while SNAPz has a high positioning error. The disadvantage of the SNAPft-z is that they computed the results on the PC, due to high calculation load. [LMP11]
Figure 2.4 shows that SNAP has a worse accuracy. It is about 1 m larger than the other algorithms in Airplace.

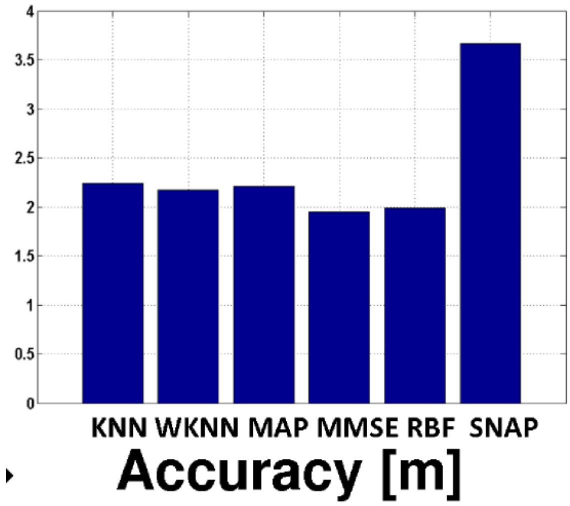

Figure 2.4: Comparison of different algorithms used by Airplace [CL11].

2.2.6 Method of Isolines

The method of using isolines has been developed at University of Applied Sciences and Art Dortmund. This method is based on Delaunay triangulation with interpolation. The method is using a network of triangles, which is created with the Delaunay-Triangulation in the area, where the positioning will be done. Each calibration point is a corner of the triangle. Within a triangle the signal strength layer can be determined by using interpolation.
The given measured signal strength of several APs can be used to determined the intersections of the isolines. This is carried out by using the calibration data. By using the measured signal strength values of several APs at a specific location, the triangles which contains isolines with this signal strength are searched. In this list a ranking of triangles by the number of intersections is created. Using triangles with a small edge length the current position can be determined by taking the centroid. Another possibility is to take the median of the centroid of the n best triangles.
The outcome of this paper is that the positioning error of this method is much better, than Bayesian and nearest neighbour method. [Röh09, GSHD08]

2.3 Inertial Navigation

An Inertial Navigation System (INS) uses data from accelerometers (linear motion sensor) and gyroscopes (angular motion sensors) to determine the current position. This system is aware of the initial position and predicts to the next position. [MW13]
The INS can be divided into a sensor part, the Inertial Measurement Unit (IMU), and a computation part which consists of a sensor model, a gravity model and mechanisation equations. [DANS+10]
INS are 'sourceless' in that they do not rely on any external transmitted signals. This explains their great utility in high-end land, air, marine and space guidance, navigation and control systems, where depending on external signals for aiding purposes might be

impractical or risky. Unfortunately, for pedestrian navigation, unaided traditional INSs are of limited use. If the upper limit to the position error is set to some reasonable value, say a few metres after some 10 s of minutes of self-contained navigation, either a very accurate navigation-grade INS or very frequent zero velocity updates (ZUPTs) with a tactical-grade system are required. [Bea09]
Figure 2.5 shows the workflow of the strapdown algorithm. It shows that the two sensors (accelerometer and gyroscope) are used as input. The data from both sensors are transformed from the body frame system into the earth fixed coordinate system. The system is first calculating the initial attitude of the device. The navigation computer is firstly calculating an initial velocity and position. The output of the navigation computer is used in the next step to calculate the current position and velocity. This is done by using the Coriolis correction and the gravitation. This information is added to the transformed accelerator and gyroscope values. After this step the navigation computer is calculating the current position and velocity. This procedure is called strap down algorithm. The strap down algorithm is additionally calculating the current attitude. This is done in the attitude computer by using the current gyro values and the initial attitude.

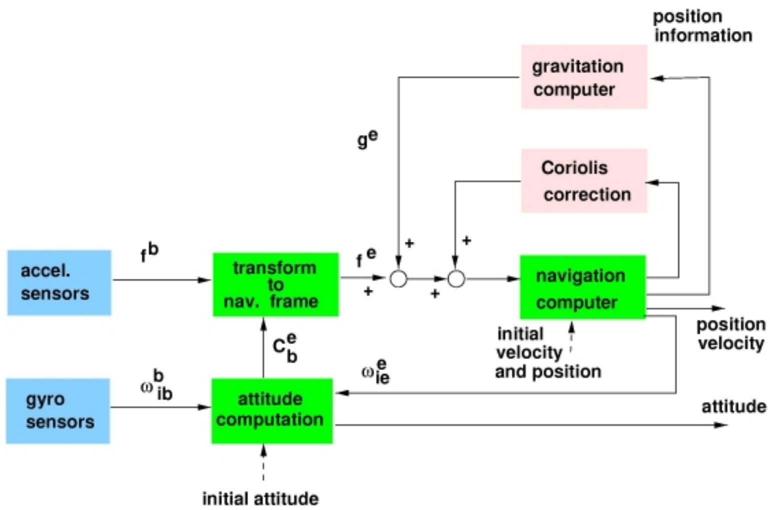

Figure 2.5: This figure show how the Strap down algorithm works. It gets the data from the motion and rotation sensors and determines the current position, velocity and attitude [Eck14].

2.4 Filter and Map matching

This section shows two types of filters: Kalman- and Particle filter. Both types are often used to adjust positions. These filter methods can be used to optimise the positioning result. It is used for optimising a probabilistic state estimation.

2.4.1 Kalman Filter

The Kalman filter [Kal60] is a process which is based on probabilistic state estimation. The filter procedure computes a prediction and uses this and the state space model to calculate a correction. The current estimation is calculated by using the Kalman weight, the measured value and the previous estimation. In Figure 2.6 the workflow of the Kalman filter is shown. In the first step a state space model is created. The next step is to iterate the data. For this a prediction is calculated. Afterwards the Kalman weight is calculated in the update step. The current estimation is based on the previous estimation and is computed in this step. A special Kalman filter for fingerprints is shown in [SALP09]. This filter is called Fingerprint Kalman filter (FKF). The FKF uses fingerprints with the Kalman filter to improve the positioning result. The method is based on Best Linear Unbiased Estimation (BLUE) approach, which is used in several Kalman variations. The prediction step is done by calculating the linear transfer function. The calculation of the weight of the calculation points and the covariance is done by using the known values in the covariance matrix of the margin of the error. The algorithm is shown in Figure 2.7.

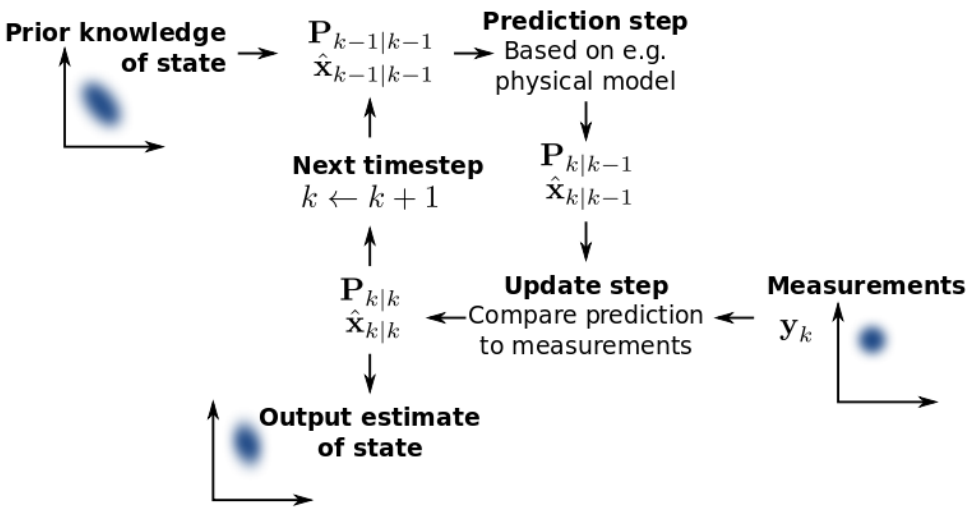

Figure 2.6: Overview about the Kalman filter [Aim14].

2.4.2 Particle Filter

A particle filter tries to estimate the state space from the actual, but unknown probability distribution. With this it is possible to deduce the possible system state of the dynamic system. The filter creates a specific number of particles. These particles are distributed in the state space. Afterwards the weight and state are calculated for each particle. The cluster represents the probability density in the start state. After each update the states of the particles are recalculated. This filters noise and the particle cloud is positioned mostly at the position where the mobile device is at that moment. In [IK12, BC09, RVA07] some particle filter methods are described. These methods are for handling indoor positioning. In [TFBD01] the Monte-Carlo method for efficient position estimation for mobile robots is shown. A special adapted version for WiFi-fingerprints is shown in [IK12]. A detailed

Algorithm 1 Fingerprint Kalman filter

1) Start with the initial estimate $\hat{x}_0 = \mathrm{E}(\mathrm{x}_0)$ and the covariance $\mathrm{P}_0 = \mathrm{V}(\mathrm{x}_0)$ of the estimation error. Set $k = 1$.
2) The prior estimate of state at t_k is

$$\hat{x}_k^- = f_{k-1}(\hat{x}_{k-1}),$$

and the covariance of the prior estimation error is

$$\mathrm{P}_k^- = \mathrm{F}_{k-1}\mathrm{P}_{k-1}\mathrm{F}_{k-1}^T + \mathrm{Q}_{k-1},$$

where

$$\mathrm{F}_{k-1} = f'_{k-1}(\hat{x}_{k-1}). \quad (6)$$

3) The posterior estimate of the state at t_k is

$$\hat{x}_k = \hat{x}_k^- + \mathrm{P}_{xy_k}\mathrm{P}_{yy_k}^{-1}(y_k - \hat{y}_k)$$

and the covariance of the posterior estimation error is

$$\mathrm{P}_k = \mathrm{P}_{xx_k} - \mathrm{P}_{xy_k}\mathrm{P}_{yy_k}^{-1}\mathrm{P}_{xy_k}^T,$$

where $\hat{y}_k = \sum_{i \in I_{\mathcal{F}}} \beta_{i,k}\bar{a}_i$,

$$\mathrm{P}_{xx_k} = \sum_{i \in I_{\mathcal{F}}} \beta_{i,k}\left(\mathrm{P}_{p_i} + (p_i - \hat{p}_k)(p_i - \hat{p}_k)^T\right),$$

$$\mathrm{P}_{xy_k} = \sum_{i \in I_{\mathcal{F}}} \beta_{i,k}(p_i - \hat{p}_k)(\bar{a}_i - \hat{y}_k)^T,$$

$$\mathrm{P}_{yy_k} = \sum_{i \in I_{\mathcal{F}}} \beta_{i,k}\left(\mathrm{P}_{a_i} + (\bar{a}_i - \hat{y}_k)(\bar{a}_i - \hat{y}_k)^T\right).$$

and $\hat{p}_k = \sum_{i \in I_{\mathcal{F}}} \beta_{i,k}p_i$.

4) Increment k and repeat from 2.

Figure 2.7: Fingerprint Kalman filter algorithm [SALP09].

description shows how the observer model has to be modeled. It also presents how the likelihood function is calculated for the observation. The model is using a radiomap. The paper is based on the three components:

- *Motion model:* Probability of motion or standstill.
- *Edge transition model:* Probability of state transition.
- *Motion distance model:* Probability of from the current position.

This algorithm is shown in Figure 2.8. The precision of the algorithm is between 200 and 400 particles in an area of 0.94 to 1.2 m. Figure 2.9 obtains the run of the particle filter in a real example.

2.4.3 Map matching

Map-Matching (MM) is used to fit an estimated path into these maps. It is often realised based on mathematical tools like the Sequential Monte Carlo (SMC) method, also referred as a particle filter: A large number of SMC particles are distributed over the digital map where rooms are represented as impenetrable walls. The method in [AKWT12] is a 2D map matching algorithm, but with reduced computation time. An achieved method is to

```
Algorithm Particle_filter( X_{t-1}, z_t )
X̂_t = X_t = {}
for m=1 to M do
  // Prediction Step
    Sample x_t^[m] with probability p(x_t | x_{t-1})
  // Update Step
  w_t^[m] = p(z_t | x_t^[m])
  X̂_t = X̂_t + <x_t^[m], w_t^[m]>
endfor

for m=1 to M do
  // Resample Step
  Draw i with probability ∝ w_t^[m]
  Add x_t^[i] to X_t
endfor
return X_t
```

Figure 2.8: Code of Particle filter, adapted to use with WiFi fingerprints [IK12].

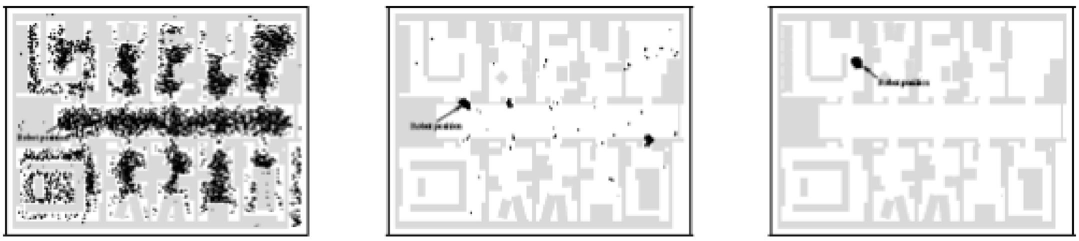

Fig. 2: Global localization: Initialization. Fig. 3: Ambiguity due to symmetry. Fig. 4: Successful localization.

Figure 2.9: This image is showing how a particle filter is working [TFBD01].

use binary weights. Furthermore the algorithm is used for 3D maps. In the paper the proposal is to use a new representation of objects in the map:

* Room: bounding walls, transitions to other objects

* Stairs: inclined planes with bounding lines and transitions to other classes

* Ladders/elevators: vertical rectangle with bounding lines

* Additional obstacles in a room (fine path planning)

For each particle some restrictions, where they can move or are not defined:

- A particle cannot move through a wall

- A particle cannot enter stairs or ladders from behind
- A particle cannot jump into the air, it can only stay next to the floor. In the case of the stairs, the position must correspond to the given height.
- A particle can leave stairs and ladders only through the transitions

For each particle and for every foot step, a collision test has to be performed. If a particle violates the constraint, then its probability will be set to zero, and it gets deleted. Otherwise the probability is set to one. [AKWT12]

2.5 Indoor Multilevel Routing

Indoor Multilevel routing is different to outdoor routing. For indoor routing a map of the building and each floor has to be available. In [HKY09] 2D-3D Spatial Database Management System (SDBMS) is introduced.
In this new 2D-3D hybrid model, the basic structure is the 2D GIS layers. CAD files for floor plans are first converted to GIS shapefiles, and then are stored into PostGIS tables. For indoor navigation, network node-links are separately built along the hallways and also stored in the database. In most routing-related applications some kind of shortest path algorithms are implemented or built-in functions are used in the application to read the memory-loaded network data. However, in the case of using a DBMS, rather than communicating with the DBMS data from the application, it is far more efficient to use DBMS-provided functions to compute the paths in terms of speed and memory use. The proposed system uses a routing method provided by the pgRouting functions. The pgRouting is a group of routing-related functions that can be used in the PostGIS database. [HKY09]
Another method for indoor routing is proposed in [SSO08]. This paper proposes to apply hierarchical graphs to indoor navigation. Hierarchical graphs can be regarded as an extension of flat graphs, in which additional layers of node and edge clusters form the levels of the hierarchy. In this paper an extraction of a base graph from a set of floor plans is shown. The topological structure of buildings, in general, is rather sparse. The typical pattern of movement would not be similar to moving from room to room arbitrarily, but more often the track will fulfil a path along main corridors or large halls before entering certain rooms. The algorithm is colouring nodes if they fulfil the following requirements [SSO08]:

- it has a connection to the exterior, or
- it represents a vertical connection such as a flight of stairs, or
- it is an articulation point in the base graph G.

The function of the algorithm can be seen in Figure 2.10.

2.6 Individual Routing

Individual routing is the possibility for the user to select the route based on individual requirements and possibilities. For path finding, a routing algorithm needs a routing graph,

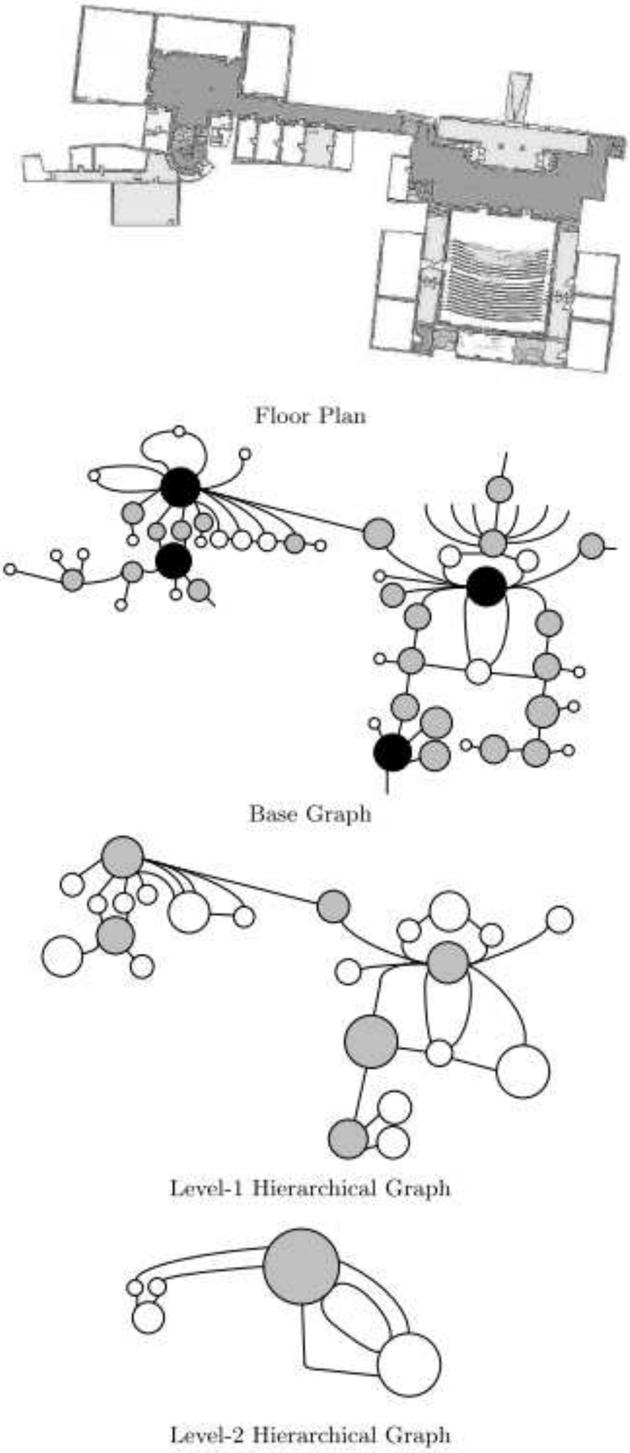

Figure 2.10: This image shows how the algorithm described in [SSO08] extracts a floor plan into a multilevel graph.

where each node connection has a weight. This weight is used to find out if this is a short way or not. The Dijkstra algorithm itself is running through all nodes in the search area until every node has been visited. It jumps always to the node with the lowest weight. The weight is calculated based on defined factors. For individual routing the weight of the nodes are changed to preference a different way. [Bau13]
With the tool OSM2po [Moe13] it is possible to convert outdoor areas of OpenStreetMap into a routing graph. In the configuration file the user can set preferred ways. These ways can be pedestrian ways, stairways or streets. The weight of these edges is set by the value set in the configuration file. This concept can be applied on every routing graph, by changing the weight of the edges. Wayfinding is a complex process and is different for individuals, depending on the purpose of the trip [XPD09]. The paper [XPD09] investigated differences in wayfinding by gender, age, type of group and familiarity with the environment. The Pearson Chi-square test was used to identify significant differences. Odds ratio, which is a measure of strength of association between two groups, was also employed. The data were generated in a case study conducted at the Koala Conservation Centre (KCC). In this paper different wayfinding strategies are used [XPD09]:

- Shortest Path: shortest pathway to navigate to the destination
- Least Time: use of least time to arrive at a destination
- Fewest Turns: move from one destination to the next in the straightest line as possible
- Most Scenic: indicates tourists preference to enjoy the scenery as they make their way from one attraction to the next
- Vegetation Type: a landmark if the vegetation is noticeably different from location to location
- Track Surface: landmarks to find the way
- First Noticed: Tourists find their way without making conscience decisions; they are responding to the external environment conditions if they use the First Noticed strategy
- Different from Previous Route Taken: strategy do so because they want to travel from site to site along an unfamiliar pathway; this adds to the enjoyment of their visit

Females tend to follow a crowd more than males, but males are more likely to use "*Vegetation Types*" and "*Track Surfaces*" as their wayfinding landmarks. Females are more likely to use wayfinding strategies such as "*Least Time*", "*First Noticed*" and "*Different from Previous Route Taken*" than males, while males prefer "*Most Scenic*" wayfinding strategies. Middle aged tourists tend to find their destination based on a "*Shortest Path*" strategy while younger tourists prefer the "*First Noticed*" wayfinding strategy. The more tourists are familiar with the environment, the less chance they will use landmarks. Tourists who are familiar with the environment are more likely than other types of tourists to navigate using the "*Shortest Path*" and "*Fewest Turns*" wayfinding strategies. [XPD09]
The outcome of the paper is that individual differences in tourist wayfinding behaviours according to gender, age, type of travel group, and familiarity with the environment.

2.7 Subjective Sense of Security

Security is a human need. Therefore people are striving for a life in secure environments. Such environments are places which are used every day. The town council of Vienna did an investigation regarding its citizens about their security feeling in public places [H0̈8]. Citizens gave some details why they feel afraid and at which places. Subway stations or territories of the drug trade are highly unsafe places for people. This investigation found out that women and older persons are more afraid about "unsafe places" than men and young people. There is a difference between unsafe places and unpleasant places.
People use their knowledge of such places and want to avoid them. Therefore they are not moving trough such places and in most cases they prefer a detour. [Reh07]
In [Reh07] research on the influence of emotions in human navigation was carried out. Both the subjective and the psychological part were investigated. Security and orientation are the most important attributes. Both factors are strongly dependable on environmental design, building structure and lighting. Humans appear not to be fully aware of environmental determinants underlying their route decision processes, although several findings suggest that atmosphere, structure, and design are of vital importance for the well-being of pedestrians. [MS08]
An article in [Rie12] is investigating the situation in Austria. The outcome of this investigation is that people want to do their day-to-day ways without disturbing by others. Such ways are the way to the underground station, the supermarket or work. Some also have the wish that no one would speak to them. The security feeling is higher if the space is free and undisturbed. Otherwise, if it is possible to make a detour around such places then the security feeling is also higher. The visibility of police officers increases the sense of security, but a multitude of police officer can cause fear. A possible solution to make some places safer is the possibility to make safety zones. In [MCRG+10] the needs of humans concerning their walking behavior are discussed. There are some theories on the needs mentioned (e.g. Maslow on the hierarchy of needs). But there are eight types of needs discussed regarding the needs of pedestrians. These types are investigated in [MCRG+10]:

- The social values and motives (contracts, relationships, transactions): interaction between different road users influences the feelings of self- assuredness. The feeling of safety is largely affected by the nature of this interaction. The possibility of participation responsibility of getting informed about social events and the kind of transport means and time needed to get there which helps to organise one's daily life (esp. for youngsters, elderly, disabled).
- Health aspects, the provided comfort (easy to walk), Avoidance of polluted air, noise, Comfort feeling by walking, Avoidance of ruthless car drivers, narrow pavements
- Comfort: The motor activities of non-motorised road users in a stricter sense are greatly affected by pavement conditions. Negative experiences are mostly caused by dirt or obstacles on the pavements.
- Weather protection (against rain, sun, snow, wind): Existence of shelter and shade.
- Safety preconditions (reflecting most of all feelings of safety): Personal security can be felt as being threatened especially if illumination is not sufficient. This is a

problem especially for women. High numbers of pedestrians tend to increase both objective and subjective sense of security because of more social control of other people in the street.

- Mobility (meaning the given possibilities to be mobile spontaneously): The mobility of women is often concentrated on environmental friendly means of transport, such as cycling. The most important feature of traffic environment from the female perspective is therefore spatial proximity. The experienced problems relate to transporting children and shopping bags. For elderly and disabled stepping over high pavements, stairs without slides or elevators, obstacles on the pavements like parked cars, garbage cans etc. can be decisive barriers, in the long run preventing people from leaving home.

- Aesthetics: This represents the desirability of stops, stations, vehicles and the environment like fresh air, calm area and green streets. But also clean areas and colours which help pedestrians or cyclists to find their way.

- Interoperability / usability: As pedestrians often are users of public transport and/or bicycles, aspects of costs and supplies for heavy bags and safe and comfortable parking areas for bicycles have to be taken into consideration. Moreover, the use of public transport will be improved by the needs of pedestrians on the way to and in the area around stops are taken more into consideration (e.g. about 70 percent of users of public means come to the stations by foot).

In [BWH+04] the subjective sense of security of young women was investigated. This paper investigated the consequence of the extension of opening hours to increase the subjective sense of security of persons at the train-station of Bern. Every person is thinking and realising their environment in a different reality. But it is possible to reuse patterns which were created by studies or different users. People tend to have their own map of cities in their memory. These maps consists of areas where people feeling more comfortably and not so comfortable. In cognitive maps danger areas are specially labelled. This helps to find out dangerous zones or areas where people do not feel comfortable. Cognitive or mental maps are described in [RPJ97]. People have relations with their environment. They store good and bad feelings in relation with other persons or their environment. Bad experience in dark narrow alleys which are not satisfactorily illuminated will lead to avoidance of walking through such streets or alleys. In cognitive maps people can draw a map of the environment and layer dangerous areas in them with a special colour. This can be used to make these areas more safe.
Women and men have a different subjective sense of security. This can be divided in more groups. Older persons are more afraid than adult people. Families and children have a high security feeling. Men are less afraid than women. Therefore pedestrian navigation system should have two options to select for the user. First of all it should be selected for whom the route is for (women, men, child or family). The second option is the age. Based on this input data the navigation system finds a comfortable route for the user by including the data from the cognitive maps and the Points of Interests (POI).
Another possibility would be to make a layer (cognitive layer) which contains the unsafe 'hot spots' and show them to the user if he or she activates this layer. The cognitive layer

has an influence on the way finding and avoiding these hot spots, if the layer is selected. Hence, there are ideas to use these inputs in navigation systems for pedestrians. As seen in [BWH+04] women feel more comfortable if they are moving through a train-station where supermarkets have extended opening hours. The same should apply to shopping streets. Most shopping centres or shops have their opening hours on a homepage. This information can be read easily and used in Points of Interest layers. This layer must be kept up-to-date. Maps in general have to be kept up-to-date. Users should be able to add information to the cognitive map layer. If many persons report the space and unsafe feeling there, then it should be added to the layer. A layer like this can help the city to improve security. Security guards can use this information in their watch routes to make them safer. Illumination is an important factor for pedestrians to find their way by night. Poor or scanty illumination makes afraid persons avoid using ways or alleys which are dissatisfied illuminated. Other factors are busy streets,squares or public spaces. People feel more comfortable in such areas, because there are other persons. These places are probably not as busy as during the day. Places like these can be safe during the day, but not at night. Then such places are becoming hot spots and people are trying to avoid this. Some of these spaces probably have beggar, drug addicts or drunks. This are factors which have an influence on the sense of security of pedestrians. To improve the sense of security, some cities have installed Closed Circuit Television (CCTV). It has increased the subjective sense of security and such spaces have become more busy. [MF08]

2.8 Summary

This chapter gave an overview about the fundamentals of positioning and routing. The first part of this chapter has shown different approaches to positioning by using WiFi and Bluetooth. The different approaches have their advantages and disadvantages. While "Neighbourhood Recognition" has a large positioning error, "Multilateration" depends on the intersection of the circles. "Distance Measuring" needs a precise timewise synchronisation, which depends on the used hardware. The "Radiomap" and "Fingerprinting" approaches can be implemented fast and produce useful results. The client based and Server based approach depend on whether the evaluation of the position is done on the device or in the infrastructure. The "Cell-based" technique is used for Bluetooth positioning, to figure out if a device is in a cell.

"Geomagnetism" is a good technology, but it can be difficult to handle and to implement. The use of "Acoustics" to do positioning have only been tested with standard hardware, while mobile phone support is restricted and has not been tested yet. Therefore, these technologies will not be used in this thesis.

This chapter has shown a few positioning algorithms. Because no better accuracy can be expected by implementing the "RBF-based Positioning Method" and "SNAP", the implemented algorithms in Airplace will be extended. Therefore, the focus is to add multifloor detection and four direction evaluation. Furthermore, a combination with inertial navigation to improve positioning accuracy will be implemented. The inertial navigation will be combined with a filter method, to get a more accurate result.

There are three different approaches of routing shown. Multilevel routing will be implemented to show the way in a multifloor building. Some attributes of individual wayfinding

are discussed in this chapter. The possibility of finding different ways in a building considering these attributes will be implemented. Different aspects in wayfinding for subjective sense of security are discussed. The approach to avoid areas, which do not fulfil the requirements of the user will be included in this thesis.
A prototype which has all mentioned approaches included will be developed to improve indoor multifloor positioning and routing.

In youth we learn; in age we understand.

(Marie Freifrau von Ebner-Eschenbach)

3 Implementation for Mobile Devices

This chapter describes the implementation details of the prototype. The practical part of this thesis is to implement an indoor positioning and routing app for Android OS. The implementation is based on Airplace indoor positioning system. Airplace is an open source indoor positioning software available from the University of Cyprus [LCC+12]. The Airplace concept consists of a recording app, a distribution and processing server and a positioning framework. This software has been improved and extended by useful features and navigation aids.

JRindoorNavigation is a software package based on Java. The apps JRindoorNavigation and JRNavRoute are developed for the usage with Android OS. This software package consists of four parts, which can be used independently. The data processing in the background is dependent on every component. The JRindoorNavigation is used to record the radiomap. This app is also used to record the routing nodes and metadata. The JRDistributionServer generates the radiomap and works as upload and download server to distribute the data. The last part is the positioning framework. This framework is used to estimate the current position. The positioning framework is included in the evaluation app and can be used as a standalone solution. Besides, the evaluation app routes the way and guides the user to the chosen destination. The workflow of the software package is described in the following sections. Figure 3.1 shows the interaction of different parts of the software modules.

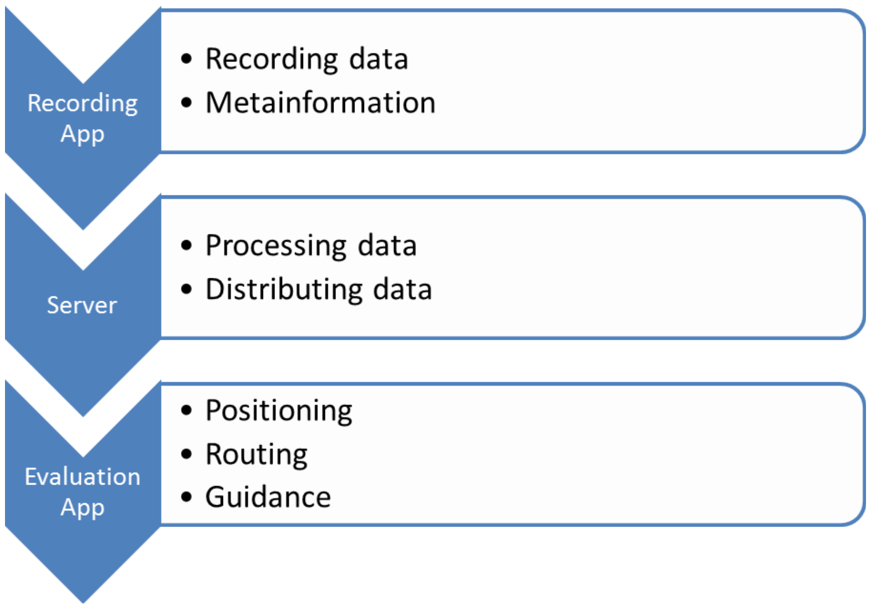

Figure 3.1: Overview about the JRindoor system.

This chapter is organised as follows: In Section 3.1 the principle of indoor positioning is described by this software package. The Section 3.2 shows the functionality of the recording app. Besides, the usage of the app is shown. The function of the server is shown in Section 3.3. The positioning framework and the algorithms are described in Section 3.4.1. Furthermore it describes the integration of the movement sensors with filters. In Section 3.4.2 the functionality of the multilevel routing is shown. The implementation of individual routing is shown in Section 3.4.3. The realisation of the avoidance of routes through closed areas is shown in Section 3.4.4.

3.1 Indoor Positioning

Airplace is delivered with four different positioning algorithms. The algorithms are described in Section 2.2. The algorithms in Airplace are implemented to determine the position. The algorithms are not able to determine the current floor in multifloor environments and the current direction. These algorithms are modified to use it with the requirements for these apps.
The extension to estimate the current floor is integrated in the same way as the position is estimated. The whole data chain has been extended in that way. Therefore the radiomap is created by considering the floor. For each position the two coordinates and the floor are stored. For estimating this data is used from the database. Based on this, in the algorithms the current floor and position are estimated. The most probable floor is taken. The four direction evaluation algorithm is implemented in the same way as the floor detection. To make all four algorithms able to determine the current position, a radiomap with recordings in four directions has to be created. The current position is estimated on the current measurements and the stored data in the database. The most probable direction is taken.
Due to variations in the electromagnetic signals the result of the floor can be wrong. Such floor changes should be avoided in order not to confuse the user. An additional feature is implemented to avoid unwanted floor changes. The feature uses the pressure sensor of the mobile phone, if a pressure sensor is available. The system stores a value after starting the app as reference value. The floor change procedure checks if the floor has been really changed. This is done by comparing the stored altitude value with the current value. These values are calculated with Formula 3.1 from [Gri14]. Hence, in this formula “alt” is the altitude. “PF” is the pressure in hPa at the flight level and “PS” is the pressure at sea level in hPa. After the level has been changed successfully the current calculated altitude value is stored as new reference value.

$$Alt = \frac{10^{\frac{\log_{10} \frac{PF}{PS}}{5.2558797}} - 1}{-6.8755856 * 10^{-6}} \tag{3.1}$$

The function used in the app is the Android built-in function *getAltitude()*. This function is available in the Android SensorManager structure and returns the current altitude in meter. This value is taken to get an usable altitude value.
The position evaluation process contains another feature to check if the current position is possible. Impossible positions are in walls or on furniture. To avoid such positions an easy method has been implemented. This method is pixel based. In the easiest implementation

the map is black and white. The algorithm will check if the current estimated position is set on a black pixel. Only white pixels are allowed for example. Therefore positions which should be set on black pixels are avoided. The algorithm is to check if a point is inside a polygon and is used to avoid positions outside the building. This implementation can be extended to allow only white pixels and disallow all other colours. Therefore estimated positions will not be displayed at places not allowed. To use this method the map has to be edited to mark areas which are not allowed in the relevant colour.
The app has an additional feature implemented, which is a compass. This compass is used to indicate the user where north is. The compass is implemented in the recording app and the evaluation app. The implementation of the compass is done, by using an accelerometer, a gyroscope and a magnetometer. The implementation to get the sensor data is shown on Android developers guide [Inc14a]. The app uses this implementation to set the compass in the right direction. The image of the compass is displayed in the ActionBar. To let the image point to north the image is rotated in the direction as the sensors values are computed.

3.2 Recording App

This app is used to record the data. The app is organised in a step by step procedure. In Figure 3.2 an overview about the steps are shown. The user is guided through the

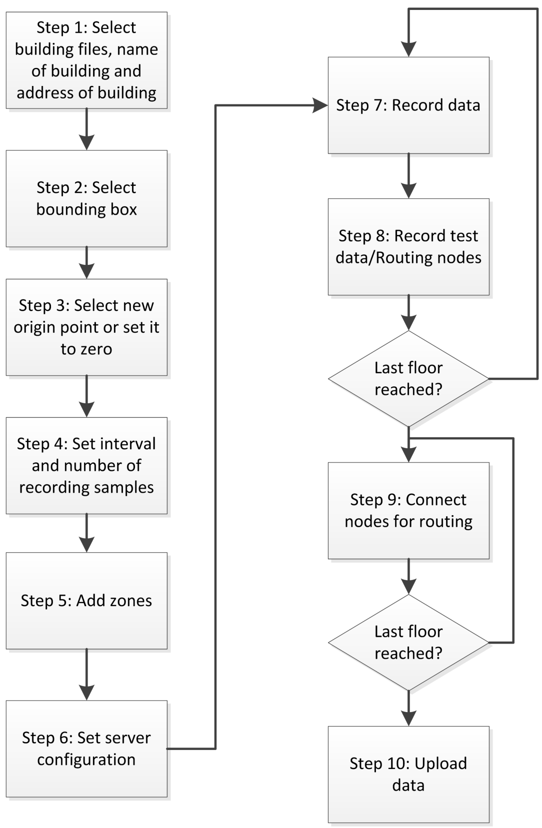

Figure 3.2: Figure shows the workflow of the recording app.

recording process. Before the app can be used to record data, the user has to provide a plan for each floor in JPEG format. The image files have to be named with the number of the floor. For instance the first floor has a file named 1.jpg and the second floor 2.jpg. Floors below the first floor (e.g. ground floor) are named with -1.jpg, 0.jpg for example. All these files are stored in the same folder on the phone. The app later uses the name of the floor plan to identify the floor in all apps. To use the number of the floor has the advantage that it can be used also for identification in data structures.
After starting the app the first step of the wizard is shown. In Figure 3.3 the first step is portrayed. The user has to select the folder, where the floor plans are stored. Afterwards a name for the building has to be selected. Besides, the address of the building should be stated. Both cannot be changed after clicking next. The system suggests some addresses by using Google Play™services. The database file is named by the name of the building. The used database is a SQLite database. Such database systems have the advantage that it can be used local without a database server. The app uses a data structure to save local used data in the memory of the smartphone. Furthermore all these data are stored in an extensible markup language (XML) file in the same folder as the floor plans. If the file *config.xml* already exists in the folder, then the settings are loaded into the data class. Otherwise a new empty file is generated and the known data are stored and loaded, respectively. At the end of each step the system stores the data in the config file. These data structures contains the name, the address and the World Geodetic System 1984 (WGS84) coordinates of the building.
The address is used to zoom the map to the building in the next step. Four balloons on the map are shown. These balloons symbolise the bounding box. In the bounding box all four corners of the building are stored in WGS84 coordinates. The app automatically calculates the width and length of the building in meters. The calculation is done by converting the WGS84 coordinates of three corners into Universal Transverse Mercator (UTM) coordinates. With this information length and width are calculated by subtracting the UTM coordinates of the two red balloons next to the blue balloon. The described step is shown in Figure 3.4.

The configuration file is shown in Listing 3.1. It contains a section 'origin point'. This section contains the offset to set a new origin point, which can be set in step three. This can be used to do recordings for the post processing module. An already set origin point can be reset by clicking the button "Set to zero". This step is shown in Figure 3.5.
Section 'numberofsamples' and 'interval' stores how much points will be recorded in which time steps. 'NumberOfSamples' contains the number of samples which are recorded. The section 'interval' stores the time which passes between the recordings. For example ten samples and an interval of 500 ms mean that ten recordings are done at a time distance of 500 ms. These settings can be done in step four. The step is shown in Figure 3.6.
In section 'zones' all possible recording zones are stored. These zones define special rooms or a general name for areas in the building. The user is able to add and remove items. The system expects at least three zones. This step is shown in Figure 3.7.
Section 'numberoffloors' is not used, but implemented for extensions. The section 'fourdirectionmode' can contain three different numbers. An explanation of the numbers is shown in Table 3.1. This setting is used to identify if the four direction recording mode is activated. This parameter is changed if the record radio map step is shown at the very

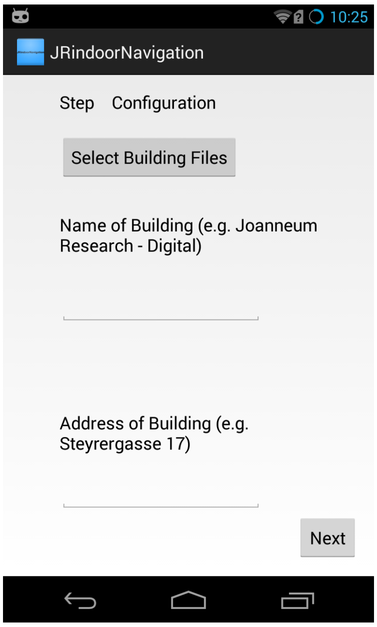

Figure 3.3: The user has to select the folder where the images of the floors are stored. The name and address of the building can be chosen by the user. Furthermore these data can be also loaded from the configuration file, if they are provided in the configuration file.

Figure 3.4: The user can move the balloons to the edges of the building. The WGS84 coordinates of each balloon are stored in the configuration file.

first time. This step is shown in 3.9.

The section ’floors’ contains all names and the path to the file location for each floor. This data is loaded in step two. The last section ’server’ contains the address and the port of the distribution server. This setting is used later to upload the recorded data. This step is shown in Figure 3.8.

The activity to record the radiomap is shown in Figure 3.10. This interface is very simple and easy to handle. The compass in the header of the app shows where north is. This helps the user to orientate in four direction recording mode to position itself to north. The first line shows the current floor and the number of detected APs. The two buttons in this line rotate the floor plan 90 degrees to the right or to the left. *RR* stands for rotate right and *RL* stands for rotate left. The next line describes this step. In this case it says “Record Radio Map”. The arrow in this line indicates in which direction the user has to rotate in four direction recording mode. The arrow rotates to 90 degrees each after

Listing 3.1: Configuration file

```
<?xml version='1.0' encoding='UTF-8' ?>
<settings>
  <name> </name>
  <address></address>
  <latitude></latitude>
  <longitude></longitude>
  <boundingbox>
    <upperleft></upperleft>
    <upperight></upperight>
    <downleft></downleft>
    <downright></downright>
  </boundingbox>
  <width></width>
  <length></length>
  <originpoint>
    <xoffset></xoffset>
    <yoffset></yoffset>
  </originpoint>
  <numberofsamples></numberofsamples>
  <interval></interval>
  <zones>
    <zone></zone>
    <zone></zone>
    <zone></zone>
  </zones>
  <numberoffloors></numberoffloors>
  <fourdirectionmode></fourdirectionmode>
  <floors>
    <floor>
      <name></name>
      <file </file>
    </floor>
  </floors>
  <server>
    <address></address>
    <port></port>
  </server>
</settings>
```

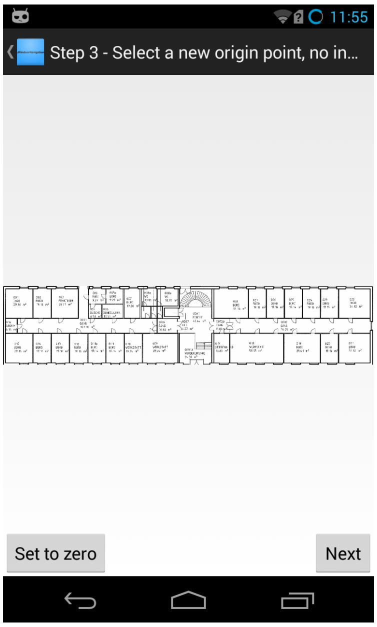

Figure 3.5: In this step the user can chose a new origin point. The new point is shown in the floor plan.

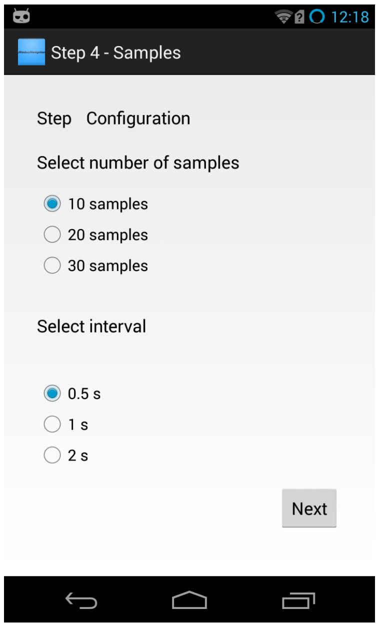

Figure 3.6: The user can chose the number of samples which should be recorded at each position. Additionally a time interval between these recordings can be set.

recording at the current position. The user has to rotate too. The line below contains the current position in x and y coordinates related to the building.
The next line has on the left side the selection of the zones. This is to set the current zone and in the right section a room number can be added, to the current position. The three buttons below are describing the actions what the user can do. To set the current location the user has to touch on the floor plan. Already recorded positions are marked red. A new position is marked light green. After a location has been set, the user can record the position by clicking the button “Record info”. A progress bar will be displayed to show the progress. For this location the entries are written into the SQLite database. The button “Next step: Record Test Data” is deactivated as long as no locations have been recorded. With this button it is possible to go to the next step. The button “Delete point” is activated, after a location is recorded. By clicking this button the delete mode is activated. The user can click any location and has to confirm the removal of the point in the database. For confirmation a dialog is shown to ask the user if he is sure to delete this entry.
The database scheme used is shown in Figure 3.16. For recording the radio map only the tables “Building” and “WiFi” are used. Building contains the data about the point. These data are the floor, roomnumber, zone as well as x and y coordinates. Furthermore

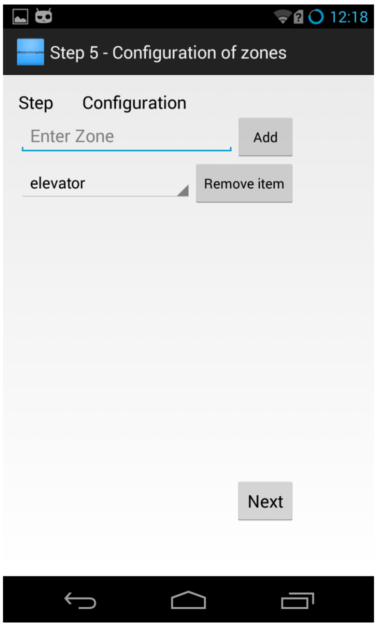

Figure 3.7: Different zones can be defined in the configuration. This is useful to assign a position to a zone. This is prior used for routing points.

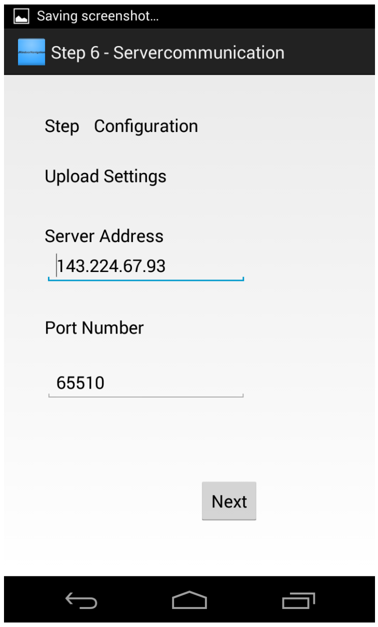

Figure 3.8: The configuration of the sever connection is set in this step.

the direction, if four direction mode is activated. Additionally for each point the offset is stored, to get it in the evaluation app. The commonid is a timestamp and used to connect the table "Building" with the table "WiFi". The table "Building" contains also the WGS84 coordinates of each point. The coordinates are calculated due to no available GPS signal in the building. The percentage of the current width or length is calculated by using UTM coordinates. The Formula 3.2 and 3.3 show how to calculate the parameters for x. For the y parameters it is almost the same. The process from converting till storing the WGS84 coordinates is shown in Figure 3.15.

$$percent_x = \frac{currentLocationMeterX}{widthOfBuilding} \tag{3.2}$$

$$WGScoordinate_x = UTMpoint_x - x\_x * percent_x - y\_xpercent_y \tag{3.3}$$

A table with the dimensions of the building and their WGS84 coordinates are stored. This is used to have the WGS84 coordinates of the building in the evaluation app. Additionally this table contains the differences between the UTM coordinates. This table is used to hand over the data from the recording app to the server and from the server to the evaluation app.

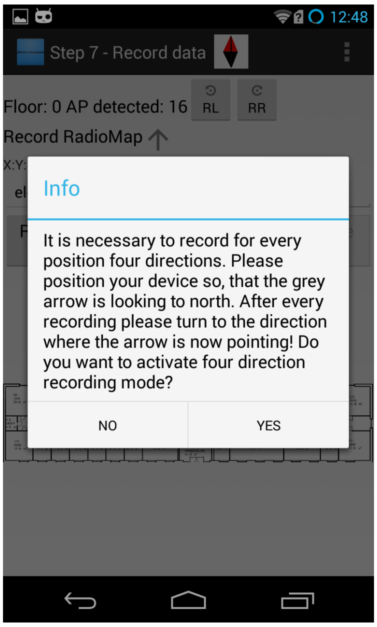

Figure 3.9: The user can chose between a normal recording and the four direction mode. This option is selected at the very first start of this step.

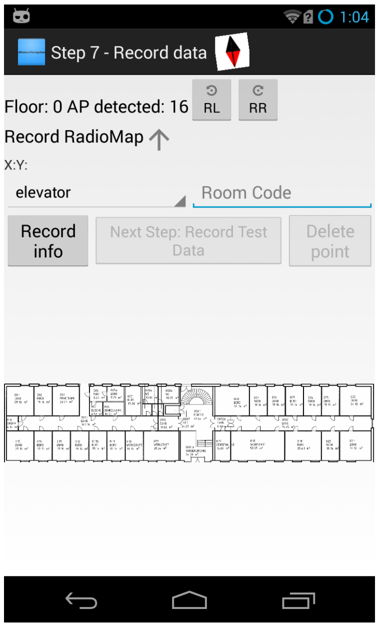

Figure 3.10: In this step the user records and manage the WiFi data.

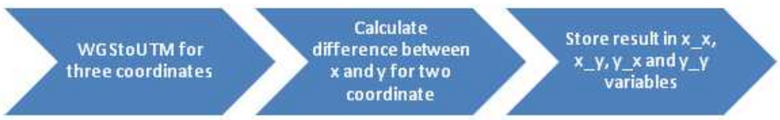

Figure 3.15: Conversion from WGS84 coordinates into UTM.

The next step is to record the test data for the same floor. These points are used for two issues. First it is used to calculate the parameters for the algorithms. With these parameters it is possible to get the estimation error and to reduce it at positioning. The second usage

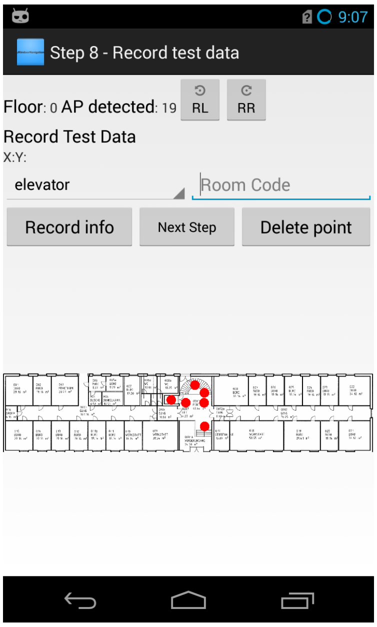

Figure 3.11: In this step the user records the points for the routing graph. Besides, additional information can be added.

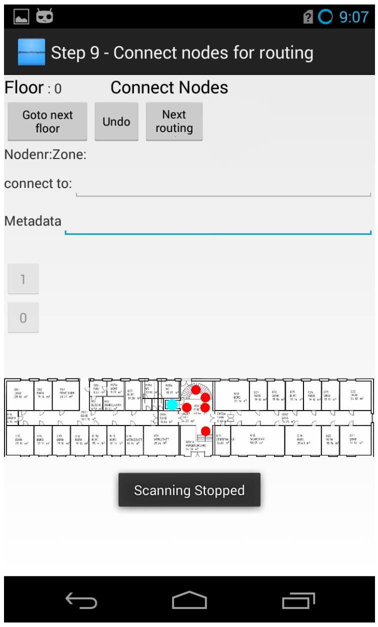

Figure 3.12: In this step the points are connected to get a routing graph. Besides metadata can be added to each point.

0	no option selected, show selection window
1	four direction recording mode is activated
2	one direction recording mode is activated

Table 3.1: The table gives an overview about the states of the four direction mode.

of these points is to use it as routing points. Therefore these points have to be positioned where the routes should be shown later in the evaluation app. The recorded data are stored in the same database. The used tables are “TestData” and “WiFi_TestData”. The structure of these tables is the same as for tables “Building” and “WiFi”. In Figure 3.11 the activity for recording test data is shown. The structure is the same as for recording the radiomap. The first line shows the number of the current floor and the detected APs. The buttons *RL* and *RR* rotate the floor plan 90 degrees to the right or to the left. The next line describes this step. In the next line the position of the current set position is displayed. This position is shown with x and y coordinates. The next line contains the properties for each position. These properties are the zones and the room code. The zone is important for routing. It defines in which area this point is. This is necessary to detect elevator and stairs later for individual routing. Further the room code is used as a routing destination. The user can set several names for the destination in this field. More than

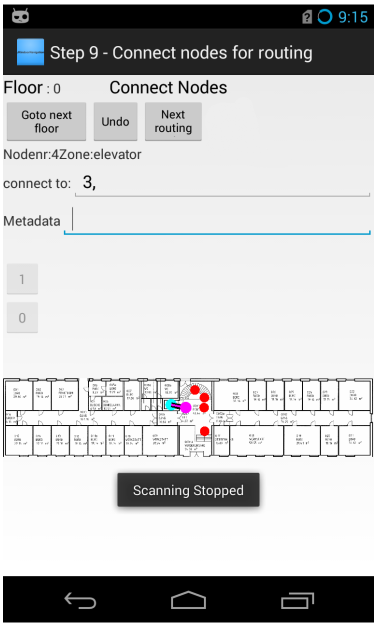

Figure 3.13: The connection of the points is done by selecting the node which should be connected.

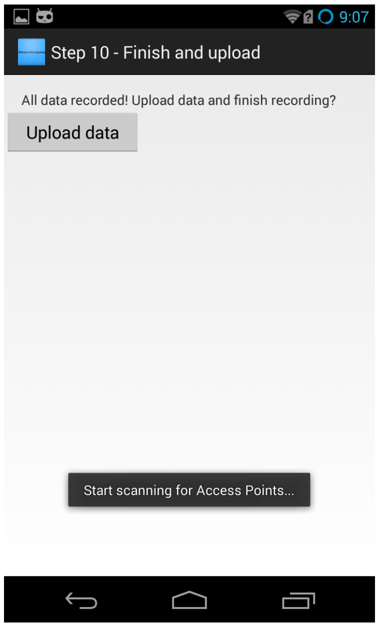

Figure 3.14: Finally in this step the user can upload the data to the processing and distribution server.

one goal have to be separated by a coma.
The button “Record info” starts to record the MAC address of the current available APs and their corresponding RSS levels. This data is written to the database. For each point the floor number, the current position in meters, the zone, a room code, WGS84 coordinates and a commonid are stored. The button “Next Step” guides the user in a multifloor building to the next floor to do the recording of the radio map. Otherwise it directs the user to the routing activity. The button “Delete point” activates the deletion mode. In this mode the user can remove points by touching the desired point and have to confirm the “Are you sure?” dialog. The currently marked point is marked black, to distinguish between the other points on the map.

The next step is to connect the routing points to generate the routing graph of the building. This step is shown in Figure 3.12.
These connections represent the possible routes through the building. The recorded routing points are loaded from the database, numbered and labelled with the floor number. The starting node is marked light blue. All others are marked red. The user has three options. It is possible to go to the next floor. This option jumps directly to the next floor. The button “Undo” removes the last added connection from the data structure. Therefore the user can correct mistakes. The button “Next routing” is to finish the connections of the current node and jumps to the next node. If the last point on this floor is connected,

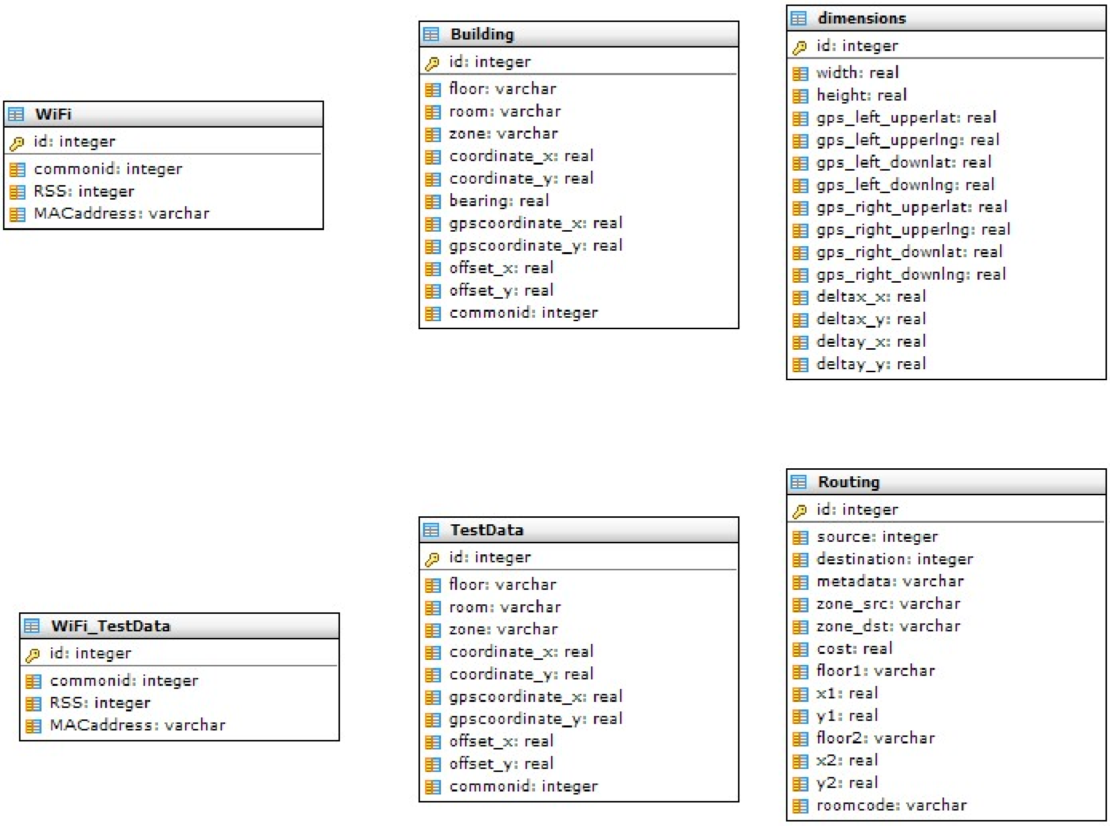

Figure 3.16: Overview about the database for the recording app.

then the activity automatically shows the next floor. In the case that the last floor is reached the activity hands over to the next activity.
The current node number and the zone are shown. Additionally the line “connect to” contains the node numbers to where this node is connected. The user can add metadata for each connection in the line metadata. This data can be used for way description. To connect the nodes the user has to touch the node(s). The system automatically adds the neighbours to both nodes. To connect the nodes one floor up or down, the user has the possibility to touch the floor number of the above or below floor. These buttons can be found on the left side of the activity. A click on this opens the floor plan of the chosen floor. In this floor the selected node is marked yellow. The user has to touch the right node and has to go back to the current floor, by clicking the floor number of the current floor. The system adds this node as neighbour to both nodes. An overview about the connections can be found in Figure 3.13.
Connected nodes are shown in magenta and connected with a magenta line to the start node. Already connected nodes from previous nodes are shown after the next routing point is shown. These nodes and lines have also the magenta colour. New connections are marked black to distinguish it from existing connections.
After all nodes have been connected the system stores the data in the database. The table used is called “Routing”. In this table the connections are stored. The weight is

calculated using the Euclidean distance. The weight of each node is actually the same. No different weighting is done. Each entry consists of following data: id, source node number, destination node number, metadata, zone of source node, zone of destination node, the cost of this edge, floor number, x and y coordinate as well as a room code. The room code is used for routing purposes to show the user the goals. The room code is related to the source of the edge.
The last step of the app is to finish and upload the data to the server. This step is shown in Figure 3.14. For this process a connection with the network of the server has to be established. The processing server must run and be able to accept connections. A click on the "Upload data" button establishes a connection to the server. The created database and the floor files are uploaded. A progress bar informs the user about the upload progress. After this step the user can close the app.

3.3 Processing and Distribution Server

The processing and distribution server is the main part of the software package. It is responsible to manage the recorded data and the floor plans as well as the processing of the data. These data are stored in a folder which is named like the database file. In this folder the upload is stored. The GUI of the server is reduced to the necessary steps. The GUI is shown in Figure 3.17. Other features from the Airplace server have been removed. The main functions of the server are generating the radiomaps and the paramters. These functions are changed to make it compatible with the used database system. The user can start/stop the server by clicking the corresponding button. This starts or stops listening on the port. In the standard case it is set to 65510. The field "Server Status" shows if the server is listening to the port or not. The user can set a threshold for the NaN value in the line "NaN value". This threshold is used to set the limit of the used RSS value. RSS value above the value are set to the threshold value. The default set value ist "-110 dbm". All RSS values above this value are discarded and will be set to this value. The white block shows the status of the progress. A click on the button "Generate Radio Map" starts the process to generate the radiomap. It processes the test data, calculates the parameters and finally packs all together with the floor plans into a zipped file.
The process itself starts with the generation of the database file named radiomap_name.sqlite.

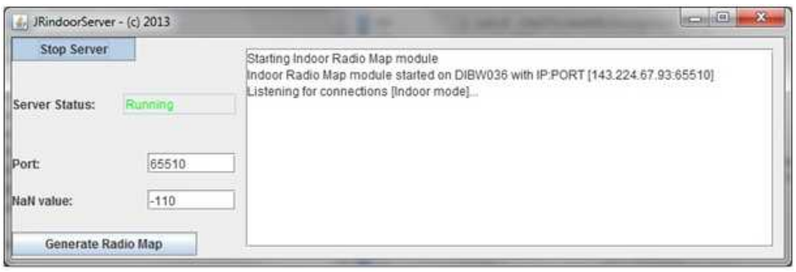

Figure 3.17: GUI of Server.

Afterwards it adds all tables to the database. In Figure 3.18 an overview about the radiomap database is shown. Two tables are just copied due to no changes of data. These tables are "dimensions" and "Routing". The other six tables are used to replicate the processed radiomap. Three different radiomaps are processed. In the table “RadioMap” for each position and each recording an entry is stored. This entry is linked with recorded RSS signals and MAC addresses in table “Wifi_signals”. The radiomap is processed by counting how much MAC addresses has been found. For each position the algorithm looks if the MAC address is recorded at this position. If it is recorded then the entry is copied to the database. If not then a new entry is created with the MAC address and the RSS value is set to the predefined NaN value. If a RSS value is below this value is also set to this value, because otherwise it is not useful.

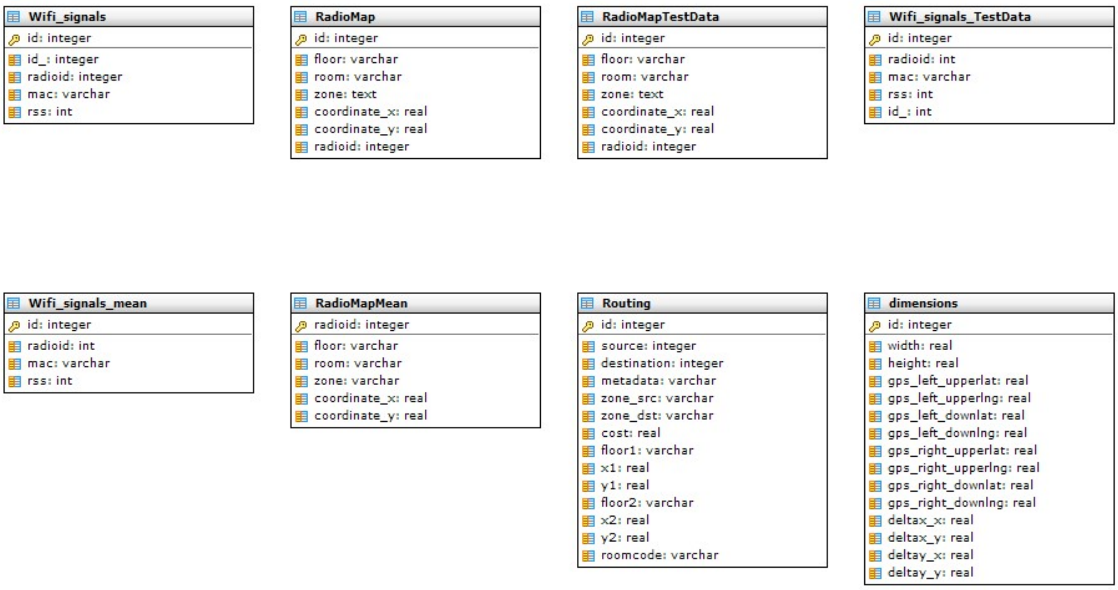

Figure 3.18: SQLite database output of the server processing.

For the mean values the algorithm iterates through the whole radiomap. In this step all the average over all recorded samples for the same position are calculated. In the “RadioMapMean” table for each position only one entry is stored. All MAC addresses and their RSS values are averaged for the current investigated position. In the table “Wifi_signals_mean” these data are stored.
The same is done for the recorded test data. But in this case only a normal radiomap is created. These data are used to calculate the average error. The error is calculated for different parameters which are used to balance some estimated positioning errors. These parameters are calculated for all four possible positioning algorithms and stored in a text file. The parameters raise the quality of the evaluation of the current position. Only the parameter which provides the lowest error is stored. The text file contains additionally the offset if a new origin point is set. Furthermore the length and width of the building are stored in there.
The last step is to put all these generated files with the floor plans into a zip file. This zip file is called like the name of the input database file. The final generated zip file is moved to the folder download. If the server is running then the evaluation app can be connected by using the IP address and the port from the server. Furthermore the name

of the building must be provided. This software provides then the generated zip file.
The recording app uses two different modes for recording the data. The “four direction” mode records samples for each direction. Therefore the process creates for one position four entries. If the “four direction” mode is detected, then columns with the name “bearing” are added to the “RadioMap” and “RadioMapMean” tables. The test data are not recorded with the four direction mode. The radiomap produces for each position an entry for each direction and for each sample. The mean radiomap generates for each positing and for each direction an entry. Therefore four entries are created in the “RadioMapMean” table. Later an evaluation of the current direction should be possible. For the calculation of the radiomap parameters only the 0°directions are used. These data are recorded in the same direction as the test data. Therefore the error can be calculated by using these data.

3.4 Evaluation App

The evaulation app is called JRNavRoute. This app directs the user through the building. This app has two steps. In the first step the user has to select the current building where he is at the moment. This is shown in Figure 3.19. In this version the user can either put in the address of the server, the port and the name of the building. With this piece of information the app can download the building files and the database from the server. The server provides a zipped file which contains the floor plans as JPEG file. Additionally it contains a text file with parameter values and it has a SQLite database file containing the radiomap. This database includes a table with the routing information.
Already downloaded building informations can be loaded, by clicking “Use existing radiomap”. Then all downloaded radiomaps are shown in a list. An example for this list is shown in Figure 3.20 The user can select the current building. A newer version of the building information always replaces the previous version.
In the next step the interface for routing and positioning is shown. How the app looks like is shown in Figure 3.21. The app has a slide menu. The overview about the menu can be found in Figure 3.22. In this menu the user can select the positioning algorithm. Additionally the routing profile can be selected. Therefore the user can choose if the route should include stairways or only the elevator. Each person can therefore select depending on their health status the preferred route. The activated option “Profile” is the route using the stairs. The deselected option uses the elevator. With the option “Guidance” the textual and text-to-speech routing information can be activated.
The user has also the possibility to select the option ‘Zone”. This option considers in order do not use nodes in an avoidance area. Therefore the routing algorithm generates a route around such areas.
In the ActionBar the user can find a compass. This compass is used to help the user to find the orientation indoors. Additional the ActionBar contains a text input box. In this input box the user can enter the desired destination of its route. The system tries to suggest several possible destinations based on the input information. The node number and name of the node are loaded from the database and stored in an *arraylist*. In the case there are several destinations available for one node, the string is split at each coma and stored in the data structure.

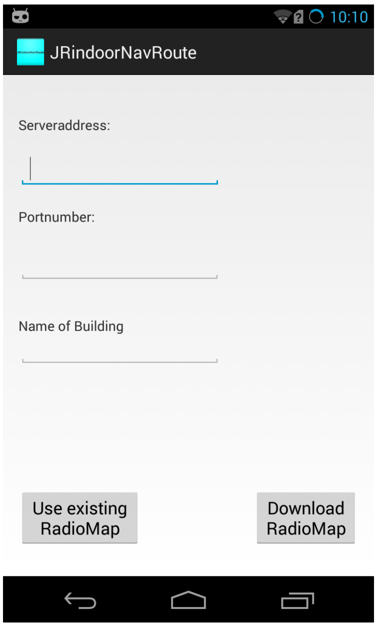

Figure 3.19: Two possibilities are available. One is to download the building data from the server. This is done by using the provided information for the server. Second is to select existing data from the app.

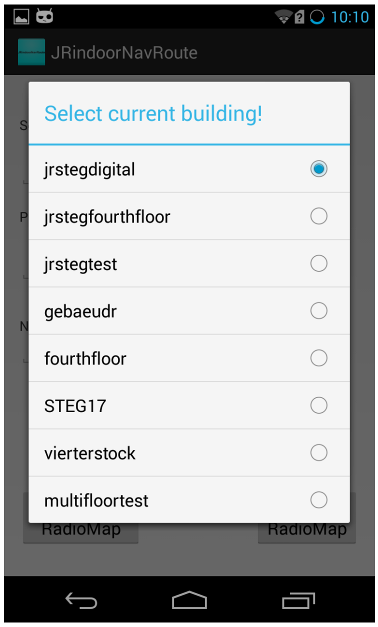

Figure 3.20: The current building can be selected in this list.

3.4.1 Real-time Localisation Framework

The real-time localisation framework is a part of the app. It is also available as a single Java program. Therefore it can be used as post-processing module. This module is used to evaluate some pre-recorded positions in a building. The inputs of the module are only WiFi recordings. The module returns the calculated position and the floor. This data is stored in a SQLite database. The results are used later to paint them on a map.
The same algorithms to determine the position as the JRNavRoute app are used by the post-processing module. The module is command line based. A radiomap and a text file containing the parameters must be provided. Both files are generated by the server. The recorded data should be provided in the JSON format in a SQLite database. The tool calculates the positions with the given information. The estimated positions are available as WGS84 coordinates. These coordinates are calculated by the in Section 3.2 described method. The results are stored in a SQLite database which has only four columns. A timestamp as identifier is given by the raw data. The second column contains the name of the used algorithm. Besides the coordinates x and y or latitude and longitude are stored,

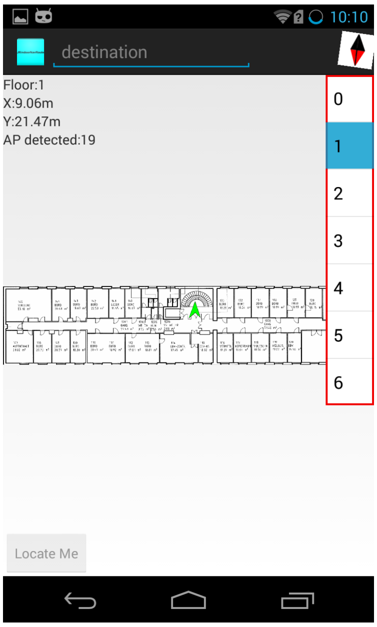

Figure 3.21: This image shows the initial state of the app. The green arrow is the current position of the user.

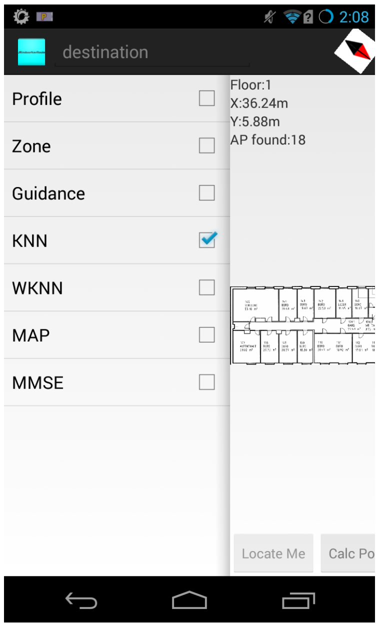

Figure 3.22: This screenshot shows the menu of the app.

respectively. The post processing module considers a given offset. This offset is subtracted from the current result and stored in the output database.
In the app the current position of the user is shown with a green arrow. This arrow points in the direction where the user is walking. To do this it is necessary to know the current angle from the gyroscope. Figure 3.23 shows the coordinate system used by the gyroscope. Based on this coordinate system the current gyroscope values are provided. Only the azimuth value is used due to the user can only move horizontal in the building. To show the current user position in the right direction an offset has to be subtracted or added. This offset is the declination of the building to north. This is calculated by sorting the WGS84 coordinates of the building by latitude in rising order. The first two coordinates are used to calculate this offset. The returned value for the compass can either be positive or negative. Therefore a negative value has to be converted into a positive. The offset is added to azimuth value of the gyroscope. The final value is used to set the current direction of the user.

The other way to compute the current direction of the user is to use the WiFi signals. This method is described in Section 2.2.3. The implementation in this thesis follows the approach in this section. The four already implemented algorithms are extended to compute the direction based on WiFi signals. For this the recording app is used to record in WiFi signals in four directions. Afterwards the server processes a radiomap. This radiomap has a recording in the mean table for each direction. In the evaluation app the current recorded WiFi signals are used to determine the current location, floor and the direction of the user. In the k-Nearest Neighbour algorithm the direction is calculated by summarising the nearest neighbours. An average of this sum is cacluated and rounded. In the weighted version the summation is done under consideration the weight. The result is used to show the current position in the map. In the Bayesisan algorithm the direction is calculated by using the probability of the current found positions from the database. The result with the highest probability is the final result.

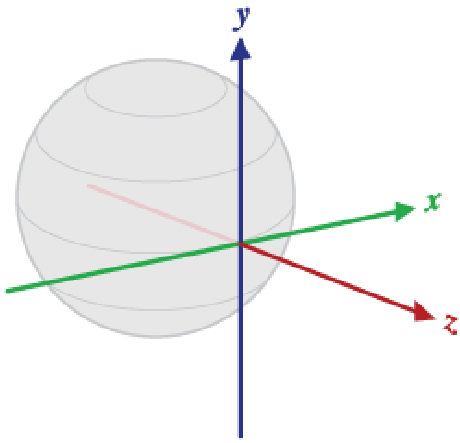

Figure 3.23: This figure shows the coordinate system of the Android OS. The used axis for the compass and the orientation of the user is the green x-axis. The used value is the azimuth value of the gyroscope. [Inc14b]

3.4.2 Multilevel Routing

The implementation of the multilevel routing is done by using the routing table from the database. This database contains the connections between the different nodes. Nodes can be connected in one floor and from one floor to another floor. Each node has a zone. This zone identifies, where a node is. This information is used for individual routing and the possibility to avoid areas which are closed. Possible routing destinations are available with a label. A label can be a room code or a name for the person to where user wants to be guided. These labels are loaded by the app and suggested to the user by using an *Autocomplete Textview*. Additionally each node has a cost. This is based on the Euclidean distance. The Figure 3.24 shows how the nodes in a floor are connected. The nodes in the elevator are connected to the node in the elevator up and/or down. The two nodes on the stairways are connected with the nodes upstairs and/or downstairs. In Figure 3.25 an overview how the nodes over several floors are connected is shown.
The routing algorithm uses the nearest node located to the current position as start node. The user can choose the destination by typing the destination in a text field and the system suggests possible destinations. If a destination is selected which is not available in the database, the app will show a dialog to inform the user about unavailable destination. The algorithm used to route the way is a standard Dijkstra algorithm. The Dijkstra is not the fastest to find the shortest path[Bau13]. But in a building there are not too much

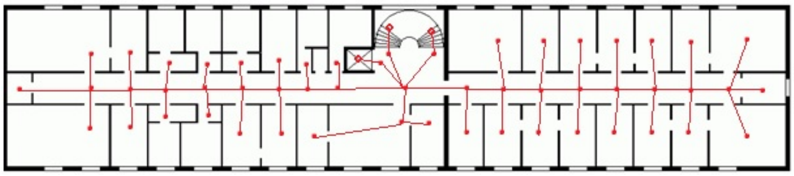

Figure 3.24: Connected nodes at a floor.

nodes to go through. Therefore the search area is limited. In such case the result is displayed very fast. In the current level only the part of the route in this floor is shown. The end of the route is marked with an arrow to show the user where to go. The direction of the arrow is calculated, by computing the angle between the two points of the last line added to the route. The Formula 3.4 is used to calculate the angle. The arrow head of the line always points in the direction where the route continues. The formula describes the calculation using the arctan of the last lines start and end points. The goal is marked with a blue circle.

$$rotation_{angle} = \arctan\frac{x_1 - x_2}{y_1 - y_2} \tag{3.4}$$

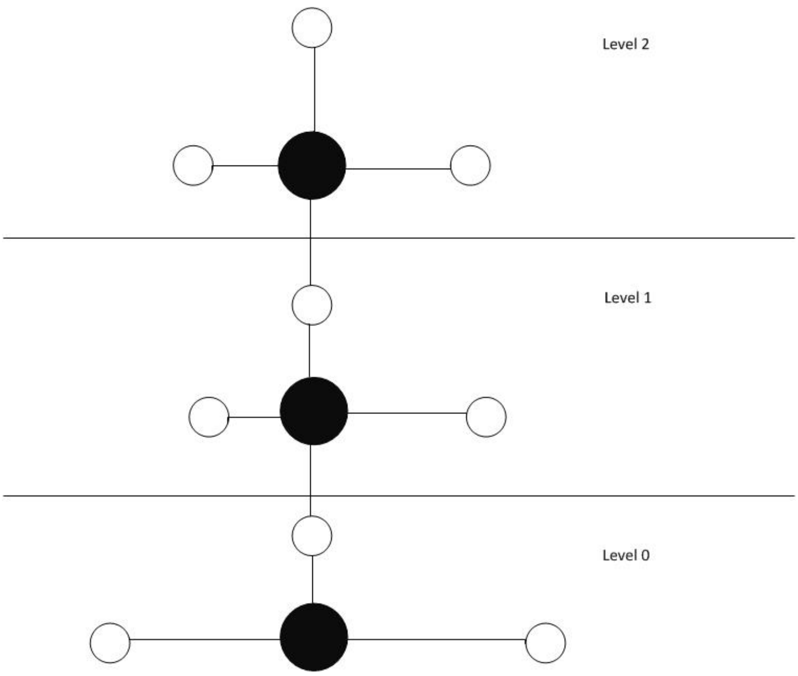

Figure 3.25: Figure shows the node connections between some levels.

3.4.3 Individual Routing

Individual routing is the possibility that the user can choose its desired way. In the case of this indoor app there are only two possibilities available. A possibility is to use exclusive stairways. The other is to use exclusively the elevator. The possibility to do the routing

only with changes in the weight of the edges is given. The problem is that the weight has to be calculated on different properties. The calculation of the weight is done in the recording app by using the Euclidean distance. This can lead to the problem, that the way in the elevator is always shorter than using the stairways.
Therefore an excluding process or nodes is used to get only the nodes which should be used. Before the routing process is started all nodes are loaded from the database. This is done the first time and if settings concerning routing are changed. While loading nodes and their neighbours the individuality is considered. This can be done by not setting neighbours which have a zone attribute named “Elevator”, “Lift” or “Aufzug”. The drawback of this method is that not added nodes cannot be used. In closed areas therefore not added nodes cannot be used for alternative routes. The other method is to change the weight of nodes which are relevant. Therefore nodes which connect nodes in the elevator have a higher weight, than nodes which connect stairs. The nodes with the zone attribute elevator are multiplied by ten and the nodes on the stairs are divided by 100. The shortest path is shown by using the stairs only. Otherwise the shortest path is the route which is only using the elevator. The advantage is that if a part of the stairs is closed, the algorithm will find an alternative route.
This solution is the best to get two different routes in a building. It helps persons to get the way what a person wants. Therefore a sporty person gets a route using the stairs. A lazy, disabled or elder person gets a route by using the elevator.
The route is shown with a line which ends up with an arrow head. The calculation of the angle is described in the previous subsection. For stairs the implementation is almost acceptable and shows a good way description. At floors which are passed by the user in the elevator only the arrow is shown. To show the user if he has to go up or down the arrow should point in the right direction. For this the algorithm calculates if the destination floor number is higher or lower than the current floor number. Therefore the arrow will point up to take the elevator to an upper floor, while the arrow will point down to take the elevator down to a lower floor.

3.4.4 Subjective Sense of Security

In Section 2.7 the background of this topic has been explained. This topic is security related and relevant for public buildings, shopping malls and public space. In this section the practical implementation of this issue is shown. It is a very important topic to route persons through buildings under exclusion of closed areas in the building. For example, a building has two stairways. The second stairway is open from 7am till 12am. After this time the user can only leave the building by using the other stairway. A person enters the building at 11am, because of a meeting in the sixth floor. The app guides the user by including the second stairway, because it is a shorter way. The meeting lasts for two hours, therefore it finishes at 1pm. At this time the stairway the person used to get to the sixth floor is already closed. The app routes the user to the selected goal by including only the still open stairway. While doing this the app checks the local database. This database contains the information shown in Figure 3.26. In the process of loading the routing graph, it considers closed areas and guides the user around such obstacles. The app also considers that persons do not start routing and start walking immediately. Therefore a time buffer of five minutes is regarded. No routes through areas which are closing in five minutes are

shown. In Listing 3.2 the code of this implementation is shown. It is not guaranteed that the user will reach the area before it is no longer accessible.
The routing graph has to consist only of nodes which are used to find a route without crossing closed areas. These areas are polygons for the system. Polygons can have several edges. Mostly used polygon areas are only polygons with four edges. These areas are marked in the map with red polygons. A polygon which blocks the way can be clicked. An alert box displays information about the area. These polygons are clickable to inform the user why this section is not passable. It is important to inform the user, because otherwise he tries to walk through this area.
To create a routing graph without nodes in closed areas, the algorithm has to check if such nodes are in a closed area. The algorithm loads all polygons from the database. The polygons are stored with the time slot where the area is open. While loading the routing graph the algorithm checks if nodes are inside the polygon. If it is true, then the node is dismissed. Otherwise it is added to the routing graph. The used algorithm to check if a node is inside or on the hull of the polygon is the *PointInPolygon* algorithm. This algorithm and applications is shown in [Bau13]. During the recording of the routing points the user cannot consider if the points later are in a closed polygon or not. The case where a routing point is not inside a polygon, but the connection of to another routing point intersects the polygon have to be considered. This is done by detect intersections of the line segment of the polygon with the routing line. The detection algorithm is shown in [Bau13]. This method makes the point in polygon detection unnecessary, because even connections to a point in the polygon intersect the polygon line segments. Therefore the algorithm is able to detect this intersection and do not add the goal node to the routing graph.
The routing algorithm finds only routes avoiding closed areas. This method is very flexible, if the database used is online. The advantage is that recent changes can flow in building the routing graph. In this thesis only a static and local database is used. Therefore the user always gets a route which is the shortest in the building under consideration of closed areas.

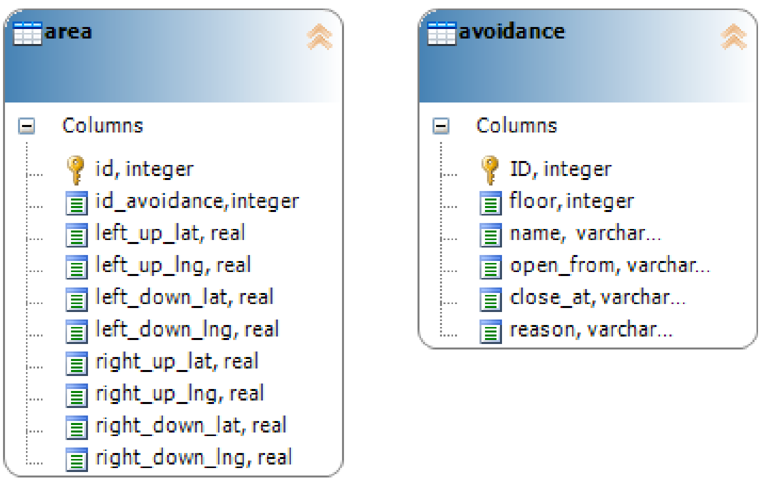

Figure 3.26: Figure shows the overview about open/closed areas.

Listing 3.2: Algorithm to decide if a node is added to the routing graph.

```
Line line_to_test = new Line(new Point(x1, y1), new Point(x2,y2));
   for(AvoidanceArea avarea: allpolygonswithinfo)
   {
      if((getCurrentTimeInMinutes()<(avarea.getOpentime()
      -tolerance_low))||
      (getCurrentTimeInMinutes()>(avarea.getClosetime()
      -tolerance_high)))
      {
      for(Line linesegment: avarea.getPolygon().getSides())
      {
         if(linesegment.getA()==line_to_test.getA())
         {
            intersection = false;
            continue;
         }
         if(!linesegement.intersect(avarea)
         {
            intersection = false;
         }
         else
         {
            intersection = true;
         }
      }
   }
}
if(!intersection)
   routinggraph.add(new Node(x1,y1));
```

3.4.5 Guidance

Navigation consists of routing and guidance. The different methods for routing are shown in the previous section. The guidance part consists of three sections. The first section is to show the route on the display. This is done by connecting the returned points from the routing algorithm in the same floor. The second part is to show a way description. This should support the user by walking the right way. The third part is a voice output. This uses the Google Text-to-Speech support and tells the user the instructions via audio. The advantage is that the user has not to watch on his smartphone permanently.
The implementation of this guidance algorithm is very simple. It handles only a few routing instructions. These instructions are straight, left and right. Besides it says the user to use the elevator or the stairs to the destination floor. In Figure 3.29 the workflow of generating the routing instructions is shown. To determine to go left or right the algorithm calculates the direction of the current nearest neighbour node and the next node. The direction is calculated by computing the angle between these two points and the y-axis. This angle is subtracted from the current direction of the position in the building. In case that the result is lower than -35 it the instruction shows “left”. If it is greater than $+35$ it shows “right”. In case that the result is in-between then the guide says “straight”. If the user reaches the destination it shows “Destination reached”. The calculation of the declination of the right direction is done by using equations 3.5 and 3.6. The add up of -90°and +360°to the result of the arctan are necessary to get a positive result and to remove the declination of 90°.

$$togo_{direction} = \arctan\frac{P_{1y} - P_{2y}}{P_{1x} - P_{2x}} - 90 + 360 \tag{3.5}$$

$$investigate_{degree} = current_{direction} - togo_{direction} \tag{3.6}$$

Angle	Instruction
< 35	Walk straight
> 180	Turn left
< 180	Turn right

Table 3.2: Table shows when the algorithm directs the user in another direction.

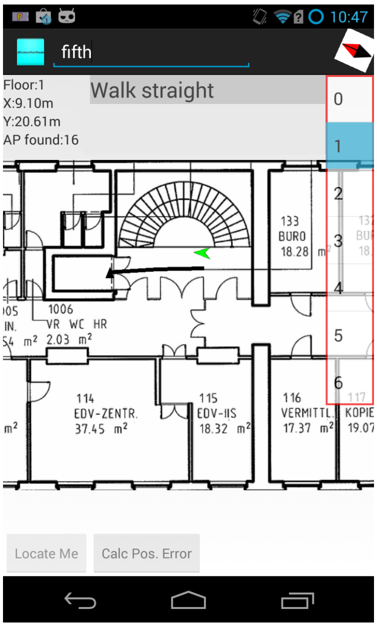

Figure 3.27: The screenshot shows the route to floor four under consideration of the usage of the elevator.

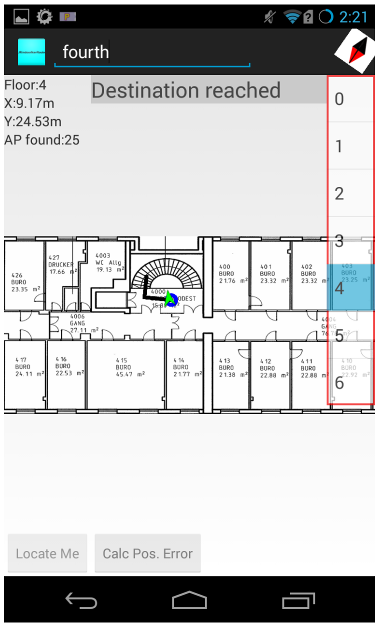

Figure 3.28: Screenshot shows the route to the destination. The guidance provides a textual interpretation of the way.

3.5 Summary

This chapter has introduced the implementation of the recording app, processing and distribution server and the evaluation app. The recording app is used to record the WiFi data on each floor of the selected building. Further test data are recorded which are used as routing points and to compute the positioning error. A table is created by the server which contains all WiFi data used for positioning. Besides a table is created which contains the data for routing. The server processes a radiomap and parameters for both, the one and four direction mode. The server provides a zipped file containing all data. These data are downloaded by the evaluation app. This app positions the user in the building. Further it can calculate the route in the building. A guidance through the building is additionally included in the app. The route is optimised to the requirements of the user. Therefore the user can chose an individual route and can avoid closed areas in the building.

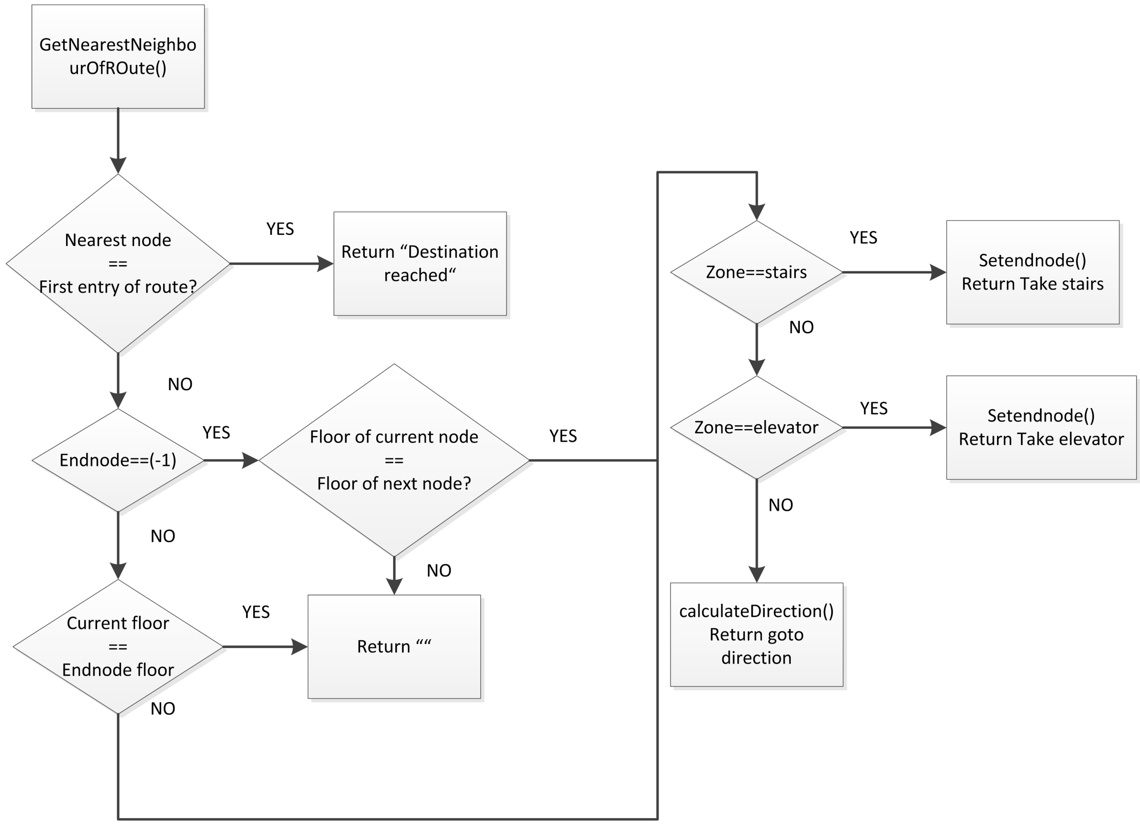

Figure 3.29: This chart shows the flow of the algorithm to generate the guiding instructions.

Effective leadership is not about making speeches or being liked; leadership is defined by results not attributes.

(Peter F. Drucker)

4 Benchmark

This chapter discusses the results of positioning and routing. It shows the positioning error of the different routing algorithms at different floors and the floor detection error. Further it also discusses the results of the implemented four direction algorithm. These results have been evaluated and compared in this chapter. In combination with inertial navigation and filters the results have been compared with WiFi only positioning. Besides this a comparison of both results is shown.
Routing has been improved with additional features. These features generate different routes and these routes have been investigated. A comparison of the different routes and results has been shown.

4.1 Test Area

The tests were done in the building at Steyrergasse 17, Graz. Tests concerning only one floor were done in the fourth floor. There are a few offices which are almost empty. Combined with the floor and the social room a suitable area was given. The test area can be seen in the Figure 4.1. Additional mobile APs were installed in the area on the fourth floor. The used APs were a Samsung Galaxy Tab 7.7 and a Samsung Nexus S (GT-I9023). Both were used as mobile hotspots and located at different locations in the test area.
Floor detecting tests were done in the staircase of each floor. This area is shown in the Figure 4.2. The blue marked rooms and areas were used in both figures to record training data, test data and to do the evaluation.

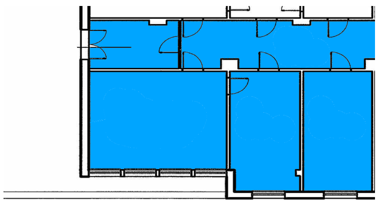

Figure 4.1: Test area in the fourth floor.

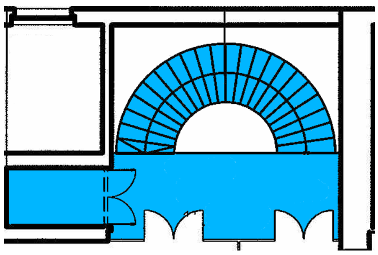

Figure 4.2: Test area "staircase" at each floor.

4.2 Positioning

This section shows three different investigations, concerning the positioning accuracy, the floor detection and the evaluation of the four direction algorithm. Firstly, it investigates the positioning error by using only WiFi signals, which have been recorded. Further it investigates the detection of the floor by each algorithm. Secondly, it investigates the positioning error by using the four direction algorithm. This should investigate positioning errors, when the user is not looking in the direction in which the data was recorded. The results are compared with one direction algorithms. The third part investigates the multisensor fusion. This is the fusion of the INS data and the WiFi data in a filter.
The Airplace system contains an error position calculation. This is used in this app for test purposes as well. The idea of calculating the error is to use the recorded test data. These data are recorded at a particular position. In the database for each of these test points the WiFi data also are recorded. The WiFi data of the recorded test data are used as an input to estimate the current position. The real position is known by the database. The positioning error is calculated by using the Formula 4.1. This error is static and only available for the corresponding recorded test data. The error calculation of Airplace has been extended to calculate wrong floors and to compute the average positioning error at each floor. Further it is extended to compute the average error between one direction evaluation and four direction evaluation.

$$pos_{err} = \sqrt{(x_{TestData} - x_{estimated})^2 + (y_{TestData} - y_{estimated})^2} \tag{4.1}$$

4.2.1 Only WiFi

This subsection shows the positioning error based on the calculated error and the general estimated declination of the error by using Formula 4.1. It is divided into three parts. The first part investigates the positioning error at each floor depending on the number of available APs. The positioning error for each floor and algorithm is calculated. The second part investigates the floor detection. The third part investigates the number of APs available in the elevator and in the front of the elevator. This should show if positioning and floor detection is possible in the elevator. The Figure 4.3 illustrates the calculation of the positioning error. The positioning error is calculated by using Euclidean distance. The inputs are the coordinates of the known position and the coordinates of the estimated position. This distance is illustrated with the black line.

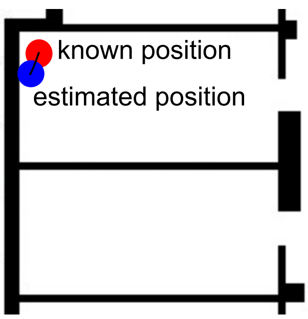

Figure 4.3: Illustrates the calculation of the position error.

Positioning Error

Different tests have been done to investigate the positioning error. Several tests were done only at one floor. A few tests were done on multiple floors. In this section the positioning error at different floors are shown. Besides this, the positioning error in the test area was calculated. This investigation considered only one direction recordings. The database consisted of a radiomap and recorded test data. The test data were recorded at specific locations. For each test position the MAC address and the corresponding WiFi signals were known. The known MAC address and WiFi signals for test location were used as input for the algorithm. The algorithm estimated the position for these signals. The result was compared with the known position. The difference of both positions results in the positioning error. Some additional influences in the positioning error have to be considered. First, the user has to set the current location while recording the data. During recording the signals at specific locations an unintentional error will be recorded. The position is set by hand on the map. Therefore the position is not accurate to cm and the position differs from the real position. This problem may influence the positioning result afterwards, but cannot be completely excluded. The only possible solution is to measure out the real position and set it on the map. But this would make the app more complicated. Therefore a small error is acceptable.
Second, the device is actually not where the set position is. In the four direction algorithm this has an influence. Using this algorithm to record the WiFi data for each position, the position is once set on the map and used for all four directions. A measurement has shown, that the set position deviates from the real standing position. The Figure 4.4 depicts that the real position of the user deviates of the device position. In the four direction mode this deviation can be up to 60 cm of the first device position. But for the evaluation this deviation has an advantage. By knowing the signals in four different directions the location can be estimated more precisely. This has the advantage that the direction is considered.
The average positioning error by using the recorded training samples is shown in Figure 4.5. The results are related to 400 recorded test samples. The k-Nearest Neighbour algorithm was not able to detect the ground floor and the sixth floor. Therefore this error has been evaluated only for the detected floors. The weighted version of k-Nearest Neighbour was not able to detect the sixth floor. This floor has not been evaluated for this algorithm. For the Bayesian algorithms all floors could be evaluated. To figure out, if the wrong floor detection influenced the positioning error in a second run these positions had been excluded from the result. These results have been shown in Figure 4.6. It shows a lower positioning error at each floor. A comparison of both diagrams clearly shows that for MAP and MMSE the results are nearly the same. The results show that the nearest neighbour evaluation is more inaccurate compared with the k-Nearest Neighbour algorithms. The results are more precise if adjusted by excluding the wrong detected errors. The average positioning error over all floors sorted by each algorithm can be found in Table 4.1. The table shows that in both cases the Bayesian algorithms are better. The best result is provided by MMSE, while WKNN has the worst results. This test has been done with one direction recording. The number of used training and test samples can be found in Table 4.2. One of the causes of the large positioning error is the widely distributed training points at the ground floor. The other reason are recorded points at the stairs.

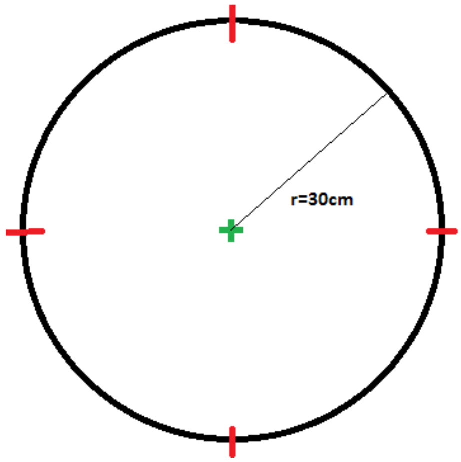

Figure 4.4: Figure shows the deviation between the user set location and the real recording location. The cross in the centre of the circle is the position of the user. The measured radius around this location is about 30 cm. The red lines crossing the circle are the positions where the device is held.

These points are not useful for positioning, because they can cause wrong floor detection and a larger positioning error as this test shows. Because of the large positioning error in this test another two tests has been executed. The test number two brought quite better results. The positioning error was between 1.41 m and 1.80 m in the adjusted results. An evaluation of the positioning error for each floor is shown in Figure 4.7 and Figure 4.8. In this test the k-Nearest Neighbour algorithms did not detect the first floor and the sixth floor. Therefore in the adjusted diagram no positioning error for floor one and six are shown. In a third test the training samples recorded at almost the same area in each floor. Therefore the same number of samples have been recorded at each floor. The number of recorded training and test data can be found in Table 4.4. The positioning error is the lowest compared with the previous tests. It is between 1.09 m and 1.66 m as shown in the Table 4.5. The evaluation for each floor is shown in the Figures 4.9 and 4.10. These figures show the positioning error for each floor. It depicts that the MMSE positioning algorithm serves the lowest positioning error. All test results clearly show that the distribution of the training points influence the positioning error. The better is the distribution, the lower is the error. Besides the error also depends on the number of training points. Lesser recorded training samples lead to a worse positioning result.

Floor detection

All four implemented algorithms were used to detect the current floor of the user. This was accomplished, by using WiFi only. There are two reasons for doing this by WiFi only. The first reason is that it is not possible to determine the current floor by altitude only. The system cannot recognise the floor by using both, but it can prevent switching the floor, if an evaluation is wrong. The second reason is that not each smartphone is delivered with an altimeter. However, such smartphones should be able to use the app as well. All four algorithms were investigated to figure out the one with the best floor

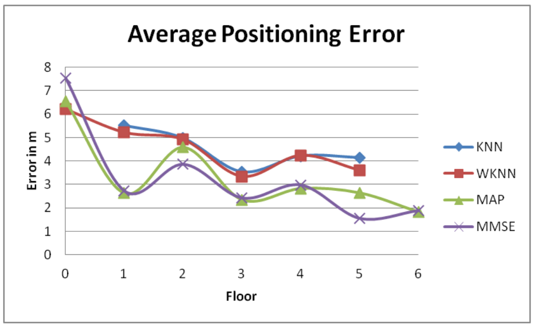

Figure 4.5: Diagram shows the average positioning error of each algorithm for the first test. The result is split over each floor.

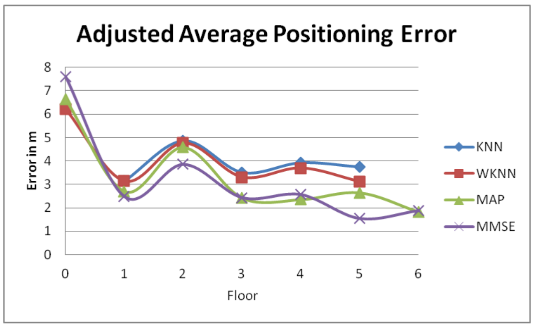

Figure 4.6: Diagram shows the adjusted average positioning error for the first test. Wrong detected floors have been excluded.

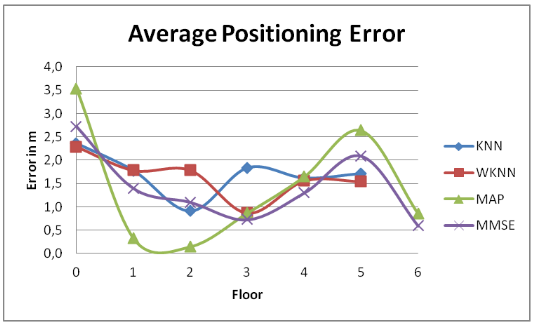

Figure 4.7: Diagram shows the average positioning error of each algorithm for the second test. The result is split over each floor.

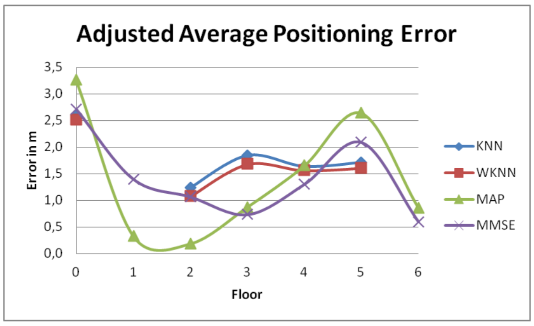

Figure 4.8: Diagram shows the adjusted average positioning error for the second test. Wrong detected floors have been excluded.

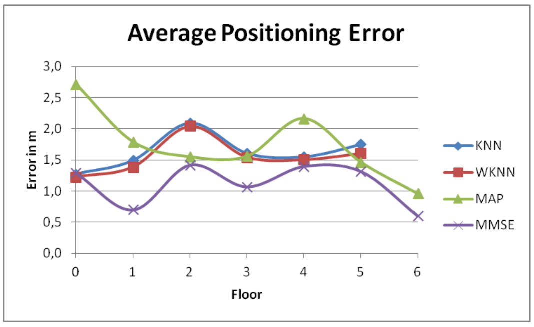

Figure 4.9: Diagram shows the average positioning error of each algorithm for the third test. The result is split over each floor.

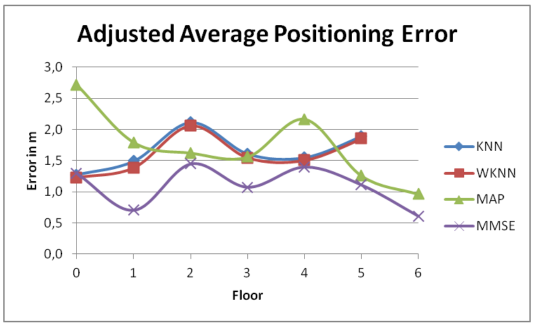

Figure 4.10: Diagram shows the adjusted average positioning error for the third test. Wrong detected floors have been excluded.

Algorithm	Avg. Error	Avg. Adjusted Error
KNN	4.48 m	3.85 m
WKNN	4.58 m	4.04 m
MAP	3.34 m	3.31 m
MMSE	3.28 m	3.20 m

Table 4.1: Table gives an overview about the average positioning errors for each algorithm used in the first test.

Algorithm	Avg. Error	Avg. Adjusted Error
KNN	1.70 m	1.80 m
WKNN	1.62 m	1.69 m
MAP	1.43 m	1.39 m
MMSE	1.42 m	1.41 m

Table 4.3: Table gives an overview about the average positioning errors for each algorithm used in the second test.

Algorithm	Avg. Error	Avg. Adjusted Error
KNN	1.63 m	1.66 m
WKNN	1.55 m	1.60 m
MAP	1.75 m	1.73 m
MMSE	1.12 m	1.09 m

Table 4.5: Table gives an overview about the average positioning errors for each algorithm used in the third test.

Floor	# Training samples	# Test samples
0	6	6
1	6	6
2	7	6
3	7	6
4	7	6
5	8	6
6	4	4

Table 4.2: Overview about recorded training and test samples at each floor.

Floor	# Training samples	# Test samples
0	19	7
1	4	3
2	4	3
3	4	3
4	4	3
5	4	3
6	4	2

Table 4.4: Table shows the number of training and test data recorded for each floor.

Floor	# Training samples	# Test samples
0	6	2
1	6	2
2	6	2
3	6	2
4	6	2
5	6	2
6	4	2

Table 4.6: Table shows the number of training and test data recorded for each floor.

detection. This test was done by using test data, which was recorded at known locations. If the determined floor corresponded to the stored floor of this test position, then it was added to the pass block. Otherwise nothing was count. In this test 400 samples were used. Set of ten samples were recorded at the same location. Therefore 40 different test data locations have been recorded, distributed over seven different floors. For this evaluation

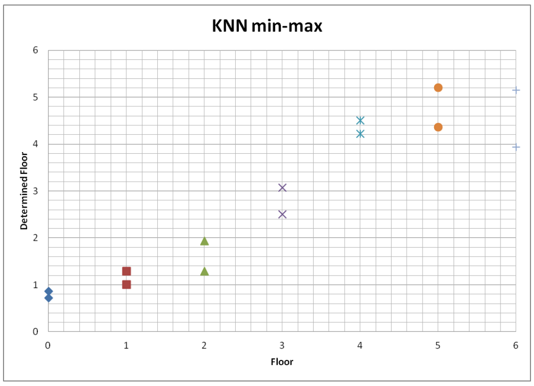

Figure 4.11: Diagram shows min and max values determined for each floor for KNN.

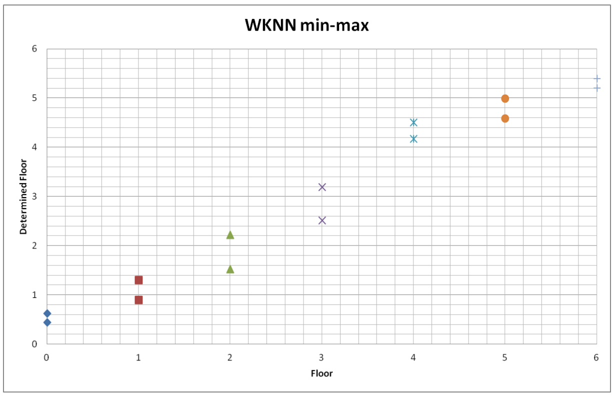

Figure 4.12: Digram shows min and max values determined for each floor for WKNN.

the same recordings as in the previous section have been used. The recorded number of test and training samples can be found in the Tables 4.2, 4.4 and 4.6, respectively. Except at floor six and the ground floor zero, all samples have been recorded at the plateau of each floor. At floor six, only the half size of the plateau was available due to architectural reasons. At the ground floor, the samples were recorded in wide spread area from the entrance to the stairs and the elevator. The diagram in Figure 4.13 compares the error hits for each algorithm. The two groups of algorithms are discussed separately. The nearest neighbour algorithms had a large error rate of about 36 percent in this test. This is due to wide spread test samples in the ground floor and the sixth floor. Positions recorded at the ground floor are counted as first floor positions. Whereas estimated positions at floor six are counted as floor five. The problem is that the evaluation of the nearest neighbour results is done by calculating the average. Therefore maybe only two or three training samples of the ground floor were nearest to the current position, while samples from floor one and two maybe also nearest neighbours. The average is then next to floor one and this is displayed as results. The weighted version of the k-Nearest Neighbour algorithm delivered a slight better result. The error rate is at 33.75 percent. This algorithm is able to detect the ground floor sometimes. For both algorithms the determined minimum and maximum for each floor number were recorded.

The diagram in Figure 4.11 gives an overview about the minimum and maximum of the k-Nearest Neighbour algorithm. The lower floors have an unique minimum and maximum range. The floors four, five and six do overlap in this range. Therefore the evaluation of this test data will switch.

The diagram in Figure 4.12 gives an overview about the minimum and maximum for this algorithm for each floor. The diagram clearly shows that the minimum and maximum range of the detected floors are not overlapping. Therefore a unique floor detection is possible. The evaluation of the algorithm does not detect the right floor, because the results are rounded. Therefore the detection failed. To get a precise result for this algorithm, the minimum and maximum range for each floor has to be stored.

The Bayesian based algorithms have a quite better result than the k-Nearest Neighbour

algorithm. These algorithms work with a probability to estimate the current floor. The results clearly show that the error rate is very low. Therefore wrong floor detections are exceptions, even with less recorded samples at each floor. The diagram in Figure 4.13 shows that the error rate of MAP is about four percent and for MMSE is about four and a half percent. Most floors have been detected correctly.
By recording enough samples at each floor, the floor detection of the nearest neighbours can be improved. A second test with 19 samples recorded distributed over the whole area of the ground floor and the same number of recorded samples at each floor as in the previous test has been done. The problem is that the weight of the samples distributed over the floors is not equal. Therefore the ground floor got a better recognition than the first floor. The evaluation of the other floors did not influence the results.

Figure 4.13: Diagram shows how often the algorithm detected the wrong floor.

Figure 4.14: Error detection after recorded three times more samples at ground floor than at all other floors.

Figure 4.15: Floor detection error by using the same number of training samples at each floor.

Due to the bad performance of the k-Nearest Neighbour algorithms, a third test has been done by recording the samples at the same area at each floor. This was necessary to figure out, if the bad results are because of less AP or because of wrong distributed samples in the area. The number of recorded samples are shown in Table 4.6. The average number of APs in each floor have been shown in the Table 4.7. It shows that in the upper floors more APs are available than in the lower floors. Therefore the evaluation of the wrong floors using k-Nearest Neighbour results from less recorded training data at the ground floor and the sixth floor. In the second test the floor detection error is a bit lower than in the first test. In this case the MMSE algorithm has the lowest fault rate, while the k-Nearest Neighbour algorithm still have the highest fault rate. The results are shown in the Figure 4.14. The third test shows an fault reduction of 50 percent compared with the first test for the "Nearest Neighbour" algorithm. The Bayesian algorithm still have almost the same error as in the first test. The evaluation of the results is shown in Figure 4.15. Besides this, the number of available APs in the staircase of the building has been investigated. The results are shown in the Table 4.7. The results in the table are filtered by different signal strength. The column < -90 excludes all recorded signals with a signal strength lower than $-90dbm$. The same is true for the columns < -89 and < -80. Only considered the column < -89 the distribution of APs is suitable. In the ground floor the number is the lowest, while in the sixth floor the largest number of APs is available.

Floor	< -90	< -89	< -80
0	17.0	14.7	4.7
1	19.3	15.0	1.7
2	18.7	16.3	2.3
3	19.7	16.3	6.7
4	20.7	17.3	5.3
5	21.0	18.7	8.0
6	21.5	20.0	10.5

Table 4.7: Overview about the average RSS in each floor at Steyrergasse 17.

Elevator Test

People may use stairs or elevator to get to the desired floor. Stairways in buildings are mostly open and not isolated by metallic walls. Therefore the WiFi signals are available for positioning. Hence, in elevator the situation is quite different, because it is a cabin made of steel. The detection of the current position in the elevator is restricted. Investigations in the elevator have shown that the average number of signal is highly reduced. The used radiomap was generated for RSS $> -90dbm$. Therefore signals lower than $-90dbm$ were discarded. The recordings were done while the door was closed. This reduced the signal strength, because of cabin are made of steel [You14]. Figure 4.16 shows a comparison of the average number of APs in front of the elevator and inside the elevator, when the door is closed. The figure shows that the number of APs is reduced by half. Only APs with a very strong signal can be detected in the elevator. This leads to positioning errors in the elevator, because the system cannot recognise the current location on account of lesser signals. When the elevator works, the recognising process is not able to detect the current location because the elevator moves faster than the detection is possible. Therefore this method cannot be used inside the elevator.

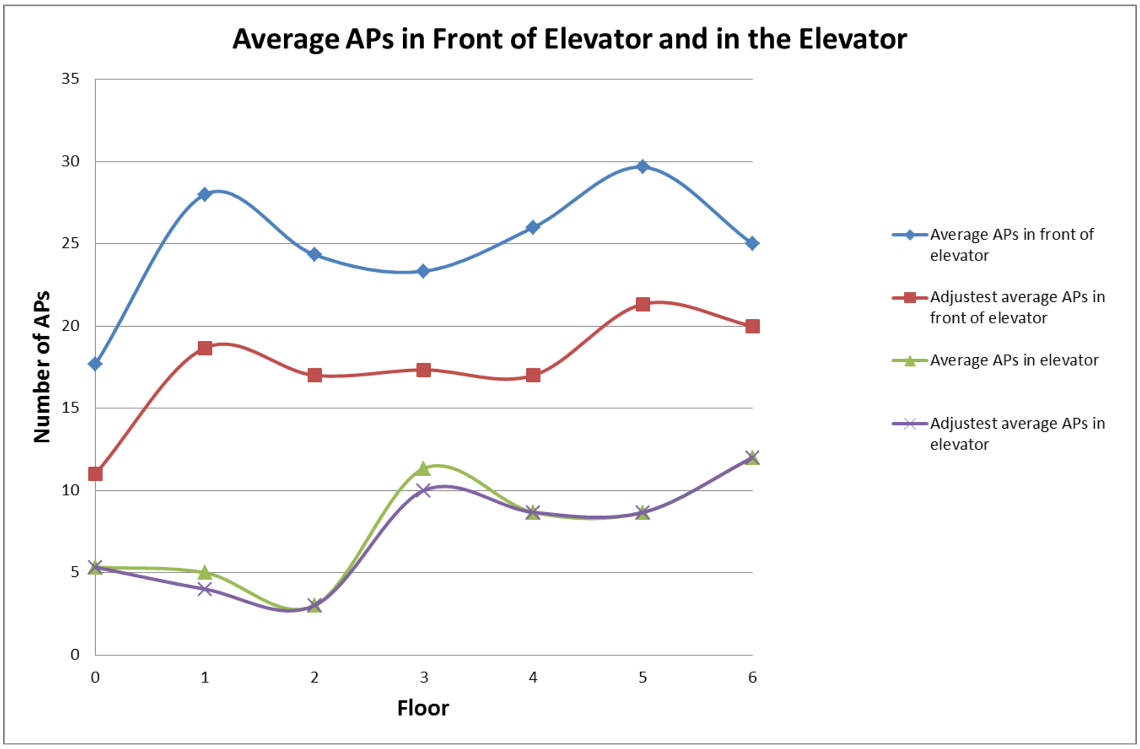

Figure 4.16: Overview about the number of average detected APs in the elevator and in front of the elevator.

4.2.2 WiFi - Four Direction Mode

This subsection shows the results of the four direction implementation. There are two expectations for this implementation: One, to determine the current direction by using WiFi signals and the second, to get a more accurate positioning result. The first part of this test investigates the evaluation of the direction. Therefore twelve test samples have been recorded in the test area. This has been done in a single floor environment. The results of the algorithm are compared with the test data. The second part of the test investigates the average positioning error of the one and four direction evaluation implementation.

Evaluation of Direction

In this part only the four direction evaluation is considered. This test evaluates the performance of the direction estimation. The training and test data recorded with the four direction mode are used. For each direction ten samples have been recorded. The evaluation is done for each sample and the average result is shown in the diagrams. The estimation of the direction is done in the same way as for the floor detection. For each direction the result is compared with the test data and evaluated.
The Figures 4.17 and 4.18 show the results of the evaluation of the direction of the k-Nearest Neighbour algorithms. The diagrams are nearly the same, because the results do not change that much. However the diagrams clearly show that the evaluation does

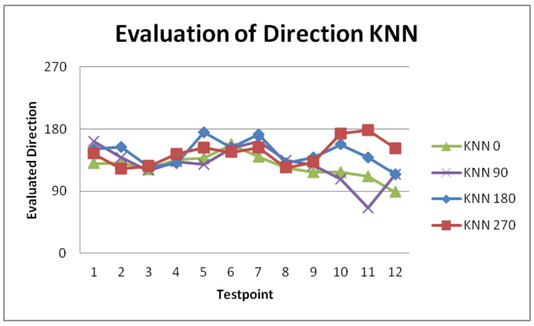

Figure 4.17: Diagram shows the evaluation of the direction of KNN algorithm.

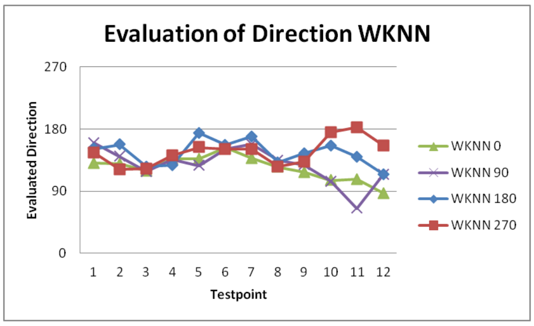

Figure 4.18: Diagram shows the evaluation of the direction of WKNN algorithm.

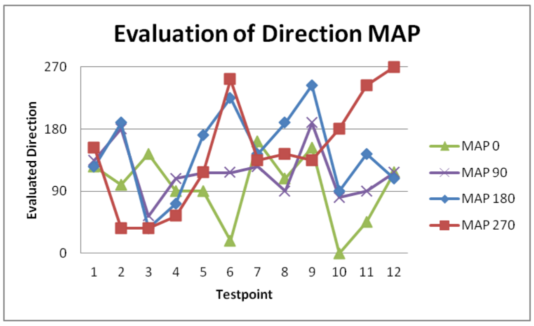

Figure 4.19: Diagram shows the evaluation of the direction of MAP algorithm.

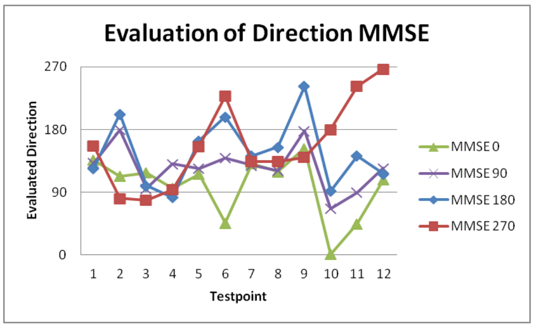

Figure 4.20: Diagram shows the evaluation of the direction of MMSE algorithm.

not come as close to the results as should be reached. The evaluation is always between 180°and 90°. Therefore, the results are not usable to detect the current direction. A possibility to get a better evaluation of the direction with this algorithm is not available.
The diagram 4.19 shows the evaluation of the MAP algorithm. This evaluation shows that estimated direction is at some testpoints where it should be. But no estimation is correct. Some testpoint estimations are completely wrong and not usable. On the one hand the wrong results depends on the current received signals, while on the other hand it depends on the number of recorded training points. The signals are almost the same if the user is in the middle of four training points. Therefore four points with four different directions are considered to calculate the direction and this direction is wrong. In average the best result has been delivered for the 90°evaluation. For all other directions only a few numbers of testpoints are near to the right direction.
The diagram 4.20 shows the evaluation of the MMSE algorithm. The results are not much better, than the MAP algorithm results. These results show the same problem as seen for the MAP results. Except some testpoints in the results are not where actual direction is. Therefore this algorithm is also not usable to determine the current direction of the user.
The results clearly show that the evaluation of the direction is not possible by using

the four direction recording mode. The estimated direction never corresponds to the real direction of the user. The floor detection results have shown that the error of the detection of the k-Nearest Neighbour algorithms is worse than the Bayesian algorithms. Therefore for this evaluation the same error rate have been expected. As described in the part of the MAP evaluation the algorithm cannot detect the direction, because of the nearest recorded training points. These training points are recorded in four directions. Mostly the nearest recordings are used to determine the floor, position and the direction. Another aspect which leads to the wrong estimations is that for the floor detection floor numbers are increasing or decreasing, while for the four direction recording only every 90°a value is recorded. As a reason of this no exact result between 0°and 90°can be determined. However, no algorithm was able to detect the direction of the recorded testpoint.

Comparison of precision with one direction recording

The implementation of the four direction mode promised to improve the accuracy of the positioning. At one position four recordings are done and therefore a more precise evaluation result is the outcome of the test. Two tests have been done. One with four direction test data and one with one direction test data. The comparison of the evaluation of four direction test data with a four direction radiomap and a one direction radiomap is shown in the Figure 4.21. The diagram indicates that the results of the four direction evaluation is quite better than the results of the one direction evaluation. A large improvement between both evaluations is shown in the KNN algorithm, while the MMSE algorithm is not that much better. The smallest positioning error has the WKNN algorithm in the four direction mode. The accuracy increased by using the four direction mode. It is more precise, because of the used training points to evaluate the current position. The comparison of the evaluation of one direction test data are shown in the Figures 4.22 and Figure 4.23. The test data are the same as previously used, but for each test point only one recording is used. In both modes for each direction an evaluation is done. The diagram for the one direction mode shows compared with the four direction mode that the four direction mode is more accurate. While in Figure 4.22 the MMSE has the best positioning error, in Figure 4.23 the k-Nearest Neighbour algorithms in three out of four cases are better. This depends on how much nearest neighbour points from the same location are used to estimate the position. Overall all seen the four direction mode delivers a more accurate positioning result. The difference between both modes is up to 0.5 m positioning error.
In comparison with the positioning accuracy of the original Airplace given in Figure 2.4 the results of the four direction mode are nearly the same. This depends on the fact, that they used own APs to do the test, while in this thesis the existing APs are used and strengthened by added two additional mobile APs. Therefore the APs in this test are used and this caused an influence on the positioning error.

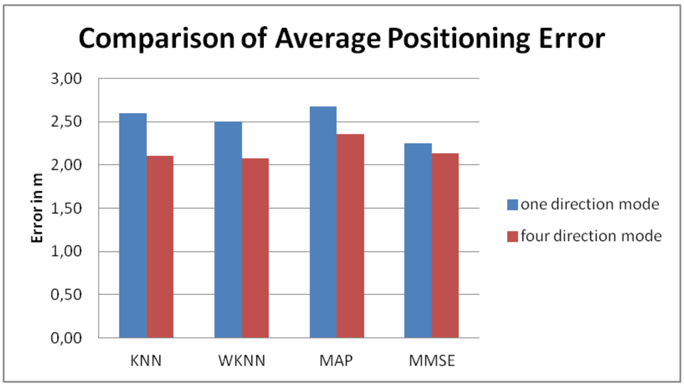

Figure 4.21: Diagram compares the positioning error by one and four direction evaluation mode.

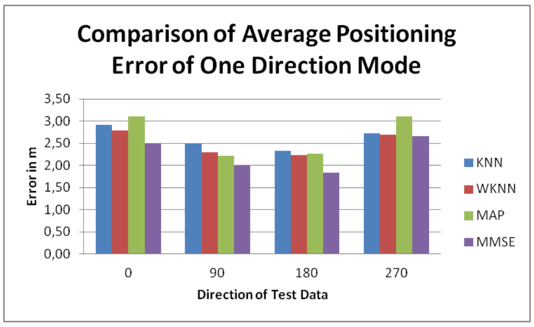

Figure 4.22: Comparison of routing algorithms with test data of different directions.

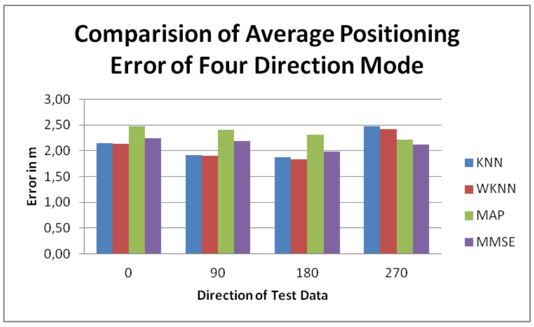

Figure 4.23: Comparison of routing algorithms with test data of different directions.

4.2.3 Multisensor Fusion

In this test the data of the accelerometer, the gyroscope and the WiFi are used to determine the current position. Dead reckoning is used to determine the current position by using the output of the accelerometer and the gyroscope. The data is recorded with a Google Nexus 5 smartphone.
The Figure 4.24 shows the track computed by using the accelerometer and gyroscope data. The green lines indicate the walls of the rooms. For this test only the social room, the large office and the small office next to it were used. The tracks are sometimes outside the walls due to inertia and the bias of the sensors. Therefore the results are not where they should be. The ground truth is shown as a black line.
In this area an additional hotspot (Hama WLAN Router MiMo 300 Express) has been installed. Two different radiomaps have been recorded in the test area, a one direction and a four direction radiomap. Both are used in the postprocessing module to estimate the position by using the recorded WiFi signals. The dead reckoning evaluation is processed in a different postprocessing module developed by Joanneum Research. In a further step the output of the postprocessing module is used in a Kalman filter developed by Joanneum

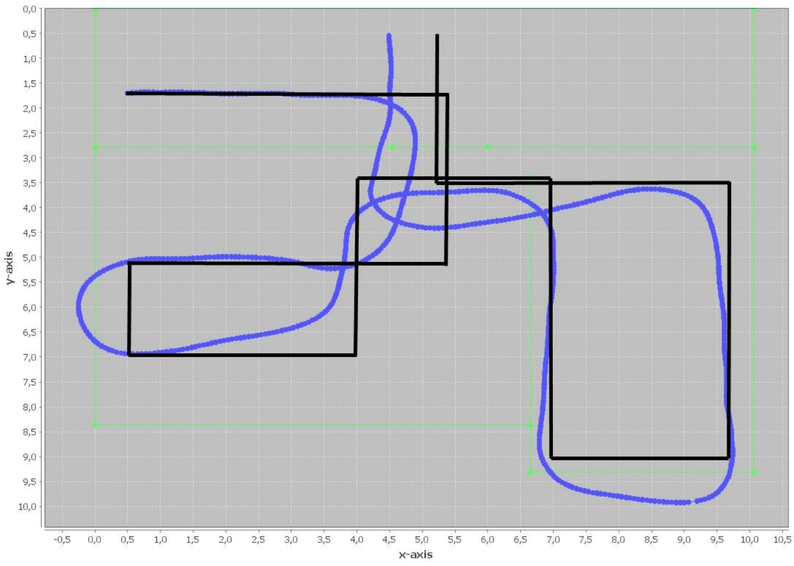

Figure 4.24: Dead reckoning track in the test area.

Research with the dead reckoning data to compute a more precise position and to filter outliers. In the Figure 4.25 the track determined by the one direction radiomap has been shown. The evaluations were done for all four algorithms and indicated in different colours. The magenta lines depict the walls of the test area. The best result was delivered by Bayesian algorithms. The startpoint is actually where it should be. The WiFi positions are recorded every three seconds, while gyroscope and accelerometer data were recorded permanently. Therefore the quality of the track does not correspond with the real walking, but it is possible to recognize the track. Outliers are in the floor above the office. These are points, where the user never walked. Another evaluation was done using the four direction radiomap. The results have been shown in the Figure 4.26. These results are more spread in the large office and actually, where the user walked. The MMSE, KNN and WKNN do have an outlier in the floor above the small office. This is the indicated start point in this evaluation.

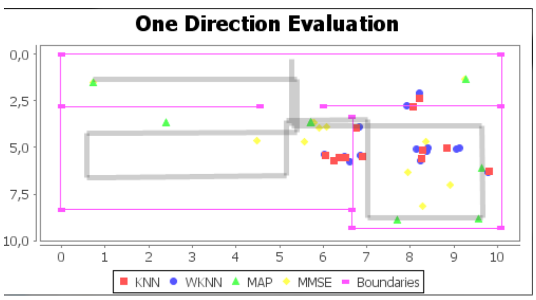

Figure 4.25: Track computed by WiFi data using the one direction radiomap.

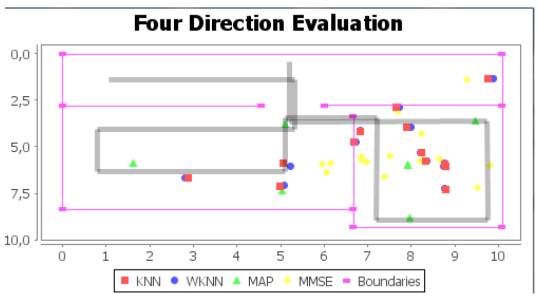

Figure 4.26: Track computed by WiFi data using the four direction radiomap.

For the filter process only the best results of the WiFi evaluation are taken. The start point of the track is determined by the first estimated point by the WiFi signals. The data fusion is processed using the WiFi data of MAP and MMSE. The grey line in the figures depicts the ground truth. The Figure 4.27 shows the filtered track of MAP and dead reckoning data. Compared with the track in the Figure 4.24 the first part of the track is completely in the room. The further course of the trajectory is nearly where the user walked. Only the part back from the small office to the floor is still not at the right position. This is caused by the WiFi data, which are positioned on a wrong place. Due to a positioning error, the WiFi position differs from the real position.
The Figure 4.28 shows the filtered track of the MMSE and dead reckoning data. This evaluation is compared with the MAP evaluation better in the area of the small office. In the last part of the track there still are the problem with the WiFi data. This ends moved track to the end, because the WiFi data does not influence the evaluation. The grey line in both figures are the real walked way.

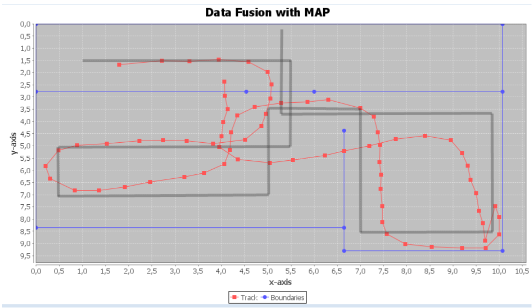

Figure 4.27: Data fusion of MAP evaluation data by using one direction radiomap.

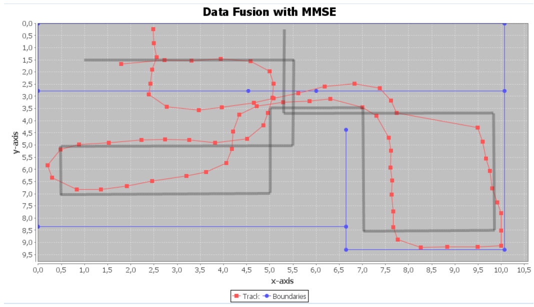

Figure 4.28: Data fusion of MMSE evaluation data by using one direction radiomap.

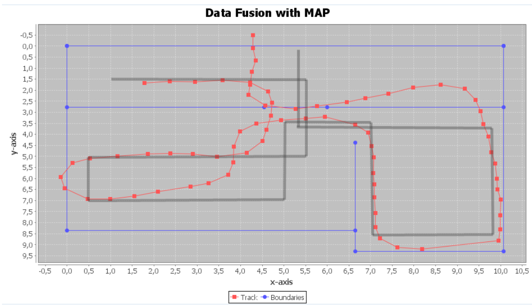

Figure 4.29: Data fusion of MAP evaluation data by using four direction radiomap.

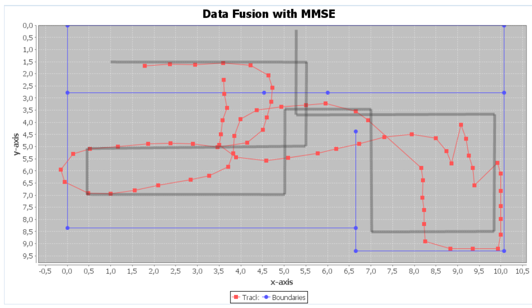

Figure 4.30: Data fusion of MMSE evaluation data by using four direction radiomap.

The four direction evaluation is shown in the Figure 4.29 for the MAP data filtered with the dead reckoning data. The four direction WiFi evaluation is more inaccurate than the one direction evaluation for the recorded WiFi data. The filter could therefore not move the track fully into the big office. In the small office the track is almost where the real

trajectory was. Only up to the last part is outside the room and differs from the real trajectory.
The MMSE evaluation of the data fusion is shown in the Figure 4.30. The first part of the track is almost the same as the MAP evaluation. In the small office there are some outliers of the WiFi positioning. Therefore the track is not as the real trajectory. The last part intersects the wall and is moved to the left side. In this part the sensor fusion was not able to correct the track.

4.3 Routing

This section shows an evaluation of the different implemented routing possibilities. It shows a comparison of using of individual routing and avoidance of closed areas in a multifloor building environment. Some examples are shown in the following section. Multilevel routing is not evaluated separately, because it is used by individual routing. There are only two different profiles available, because indoor do not offer more options. Besides, it is integrated in the part to avoid closed areas.

4.3.1 Individual Routing

Two different profiles are provided in the evaluation app. The user can choose the preferred route by activating the profile mode. The system shows the selected way through the building. The Figure 4.31 shows the route using the stairs. Only the route in the current selected floor or the current detected floor is shown. The Figure 4.32 shows the route using the elevator. Besides this, if guidance is activated the app shows routing instructions in the upper part of the display. The Figure 4.31 shows the route with instructions. The arrow at the end of the line indicates the direction where the user has to go.

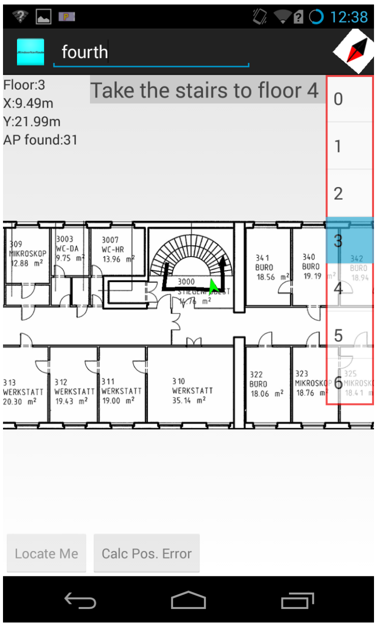

Figure 4.31: In this figure the individual route using the stairs is shown.

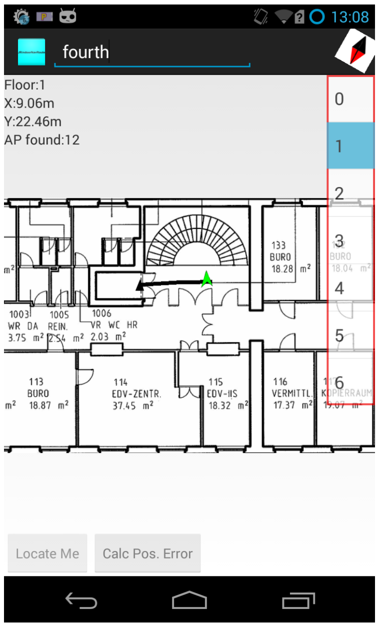

Figure 4.32: In this figure the individual route using the elevator is shown.

4.3.2 Subjective Sense of Security

The results of the implementation to avoid unsafe or closed areas are shown in this section. The results clearly show that closed areas can be avoided. The user is guided around such areas under consideration to show the shortest path by not walking through the closed area. The method has an advantage that the user is guided through the building without losing time on such areas. The information and the reason of closed areas are stored in a database. The other aspect of this kind of routing is to guide the user only through safe and allowed areas in the building. An overview on how the algorithm handles closed and non-closed areas is shown in Figures 4.33 and 4.34. In the Figure 4.33 the route shows the shortest path without considering the closed area. This way is shown either if the door is open or the user has deactivated the option “Zone”. In this case the option “Zone” is activated and the door is open. The possibility of deactivating this option is that persons can use the app, if they are eligible to access closed areas. The menu point where this option can be activated is shown in the Figure 4.36. Compared with the Figure 4.34 the route considers the closed door and finds the shortest path around this door. The door is only open between 9 am and 11 am. The user is guided around in order not to use the closed door. The information about the closed door is shown in the Figure 4.35. A click on the “closed area” which is symbolised by the red polygon shows the dialogue. The dialogue shows two important pieces of information. These are the reason of the closed area and the time in which this area is closed. This function can be activated and deactivated. The activation is done in the menu by clicking the option “Zone”.

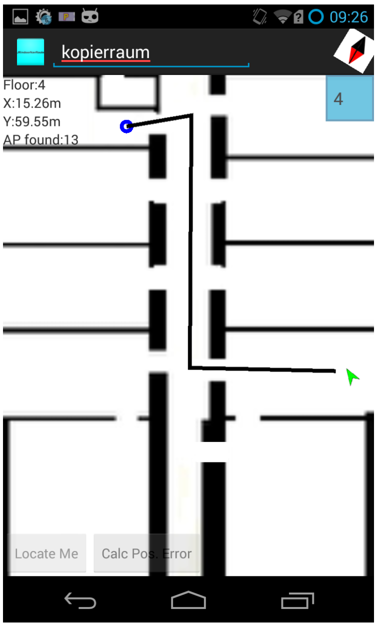

Figure 4.33: The shown route in this figure is the shortest path from current position to the destination. In this route there exists no closed area.

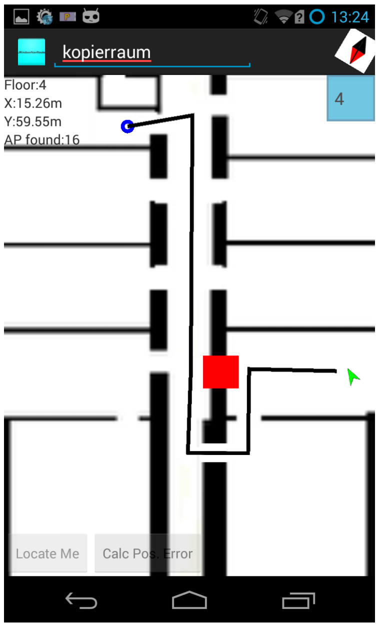

Figure 4.34: The shown route in this figure avoids the closed door. Therefore the shortest path is to use the way through the next office.

4.4 Summary

The positioning results show the positioning error concerning the position, the floor and the direction. Two different methods have been investigated. The one direction mode was used to detect the current floor. The results indicate that it depends on the number of recorded training points, the distribution of the points and the area where these points have been recorded. Further the floor detection is more precise by using the Bayesian algorithms, while the k-Nearest Neighbour algorithms have larger errors. The positioning error varies over the floors. This is because of the different number of available APs and recorded training points.
The second investigation was to figure out if the four direction recording is more precise than the one direction recording. The comparison of the four direction mode and one direction mode positioning error shows that the four direction mode is able to estimate a more precise position than the one direction mode. This is because for one position four recordings are available.
The four direction recording was used to detect the current direction of the user. The results have clearly shown that this is not possible. Therefore the gyroscope was used to determine the current direction of the user. This was more exact, while the four direction recording could only detect the position in 90°distances. The values that the gyroscope

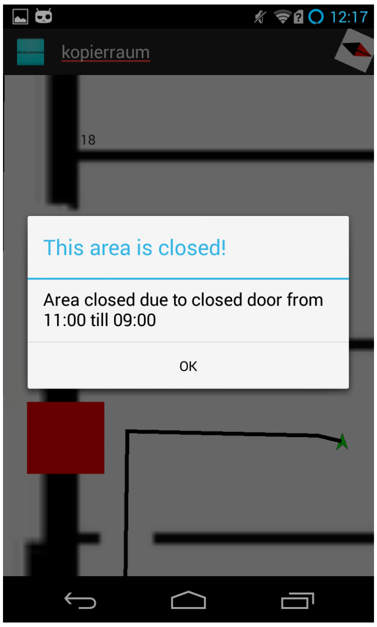

Figure 4.35: Information about the closed area.

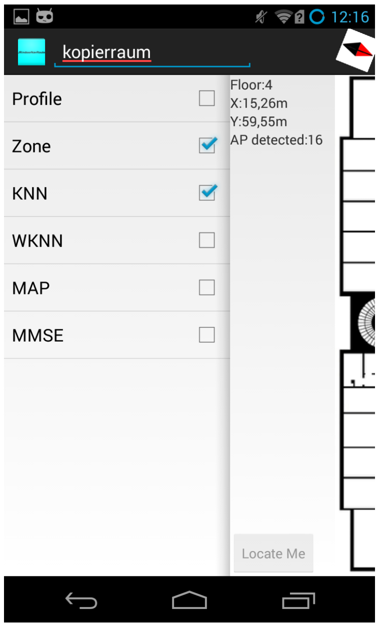

Figure 4.36: The user can activate this function by clicking an entry in the menu. If the option is activated, the route to avoid closed areas is shown.

delivers produces an error, which becomes larger over the time. An investigation about this error is shown in [SR11]. To reduce this error a calibration of the sensor should be done after starting the app. This can be done by drawing the figure “∞” with the device in the air.
The multisensor fusion shows that the track created by dead reckoning can be improved by filtering it with the WiFi data. The results shows a partially improvement of the track to the real trajectory.
The results about routing show the different possibilities of pathfinding. In this chapter individual routing has been presented. It shows the differences between routing the way using stairs and the elevator. The individuality of persons could be evaluated. The instructions of both routes are different. Another shown routing aspect is the possibility to avoid closed or unsafe areas in buildings. This improves the quality of routing. It helps the user to find the shortest route, while not using any closed sections. Besides it supports security and facility service to keep persons outside of closed sections. Summarised both routing features extend and improve the routing service for pedestrians in buildings.

Anyone who has never made a mistake has never tried anything new.

(Albert Einstein)

5 Conclusion

5.1 Summary

In this thesis the indoor positioning app system Airplace [LCC+12] has been improved. Both apps, the recording and evaluation app have been redesigned and extended in its functionality. Additionally the processing server GUI has been reduced to the necessary functions. The recording app is organised as a wizard and guides the user through the recording process. The data management has been reorganised and uses a SQLite database. This speeds up transactions and reduces storage and loading times. In a first step the recording app asks for the settings and then the data is recorded. Furthermore, the user is able to create a routing graph.
The server processes the radiomaps and copies the routing graph. Furthermore, the server manages the upload and download of the data. The processed data is stored in a zipped file, which contains all processed data and the floor plans. Therefore, the evaluation app only has to download a file, which includes all necessary files for positioning, routing and displaying the floor plan.
The evaluation app and the localisation framework are used to position the user in the building. While the localisation framework is used to do post positioning with recorded data, the app is used for localisation and routing in real-time. The positioning algorithms have been extended to detect the current floor. Furthermore, a determination of the current direction has been implemented, by using the gyroscope of the device. A routing and guidance aid has been added, to support the user to find their way.

The main focus of the thesis is to improve the positioning. In the first part all included positioning algorithms have been adapted to detect different floors by WiFi. In a multi-floor building the positioning algorithm is able to detect the floor and the position at that floor. The outcome of this implementation is that the "Bayesian" algorithms do better floor detection. They are based on the probability, while the k-Nearest Neighbour algorithms work with distances. The positioning error tests from Airplace have shown that the "Bayesian" algorithms have a lower positioning error in comparison with the k-Nearest Neighbour algorithms.
The floor detection should be available also in the elevator, but the tests have shown that this is not possible. Because of the cabin made of steel the WiFi signals are attenuated if the door is closed. Therefore the positioning in the elevator is not possible.
The implementation of the four direction mode improved the accuracy of positioning. The outcome of these tests has shown that the positioning error is about a half meter lower compared with the one direction test. Because of four recordings done at one position in all four directions the evaluation is more precise. An evaluation of the current direction was not possible, due to several recordings in the neighbourhood and varying WiFi signals.
The multisensor fusion merged the results of the WiFi positioning and the output of ac-

celerometer and gyroscope. The output has been filtered to get an accurate result, which excludes outliers. The outcome of the tests has shown that the track produced by accelerometer and gyroscope values can be improved.
Three aspects of routing have been introduced and implemented. The possibility of multifloor routing has the advantage to navigate a person in a multifloor building. Therefore the user can be guided from floor to floor. The change from a floor to another depends on the selected profile. The usage of different routing profiles in buildings has the benefit to get the desired route. The user can decide whether they want to use the stairs or the elevator. The system calculates the preferred route by using different weights for each profile. Additionally guidance is included to show the way by text instruction and text to speech. Closed doors, insufficient illumination of floors or maintenance in the staircase may cause a subjective sense of insecurity. To avoid that the user can access one of these areas, the possibility to exclude the route through such areas has been implemented. Therefore the user gets a route through an area which they are allowed to access.
This thesis has shown an improved indoor positioning and routing system, which makes it easier for users to find their way in a building.

5.2 Future Work

This thesis solves problems concerning indoor positioning and routing. It tries to improve routing and pathfinding in buildings. There are still some topics, which could be used to extend and improve the work. The app uses floor plans provided in JPEG format. The disadvantage is that these plans do not provide any further information. IndoorOSM [Ope14] is a project of OpenStreetMap which is a standardised system of mapping buildings and floors. Therefore more information about the building can be provided.
Augmented reality is a new feature becoming more popular. It shows the way in the augmented reality of the user. Therefore instead of a map the current camera image is shown on the display. In this image the direction to go is depicted by an arrow.
The implementation to get information about the building is not sufficient for a user. The usage of QR code technology, NFC tags or Bluetooth beacons makes it more comfortable to get the information about the building.
An online database which contains the information about the closed areas will be benefit the routing for the user. This database should be maintained by facility and security management. Once a radiomap have been recorded, it must be maintained. APs can be added, changed or removed, and therefore the positioning might lose precision. A mentioned, but not implemented part in Section 2.1 is the usage of updating the radiomap. In large buildings with many APs an implementation with a scoring system could help to keep the radiomap up to date.
The used filtering methods and map matching have been used in external post processing modules. To provide a more accurate positioning this must be implemented in the evaluation app.
This prototype therefore can be extended to provide the user a fully comfortable indoor navigation system. Including all described features would make a good feature rich and accurate indoor navigation app. The client server based application has the advantage that the administrator and the facility service can provide all data in a database.

Bibliography

[Aim14] Petteri Aimonen. Basic concept of kalman filtering. http://commons.wikimedia.org/wiki/File:Basic_concept_of_Kalman_filtering.svg, Feb 2014.

[AKWT12] C. Ascher, C. Kessler, R. Weis, and G.F. Trommer. Multi-floor map matching in indoor environments for mobile platforms. In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–8, 2012.

[Bau13] Clemens Bauer. *Survey and performance analysis of shortest-path routing approaches on mobile devices*. Project thesis, 2013.

[BC09] I Scowcroft J Mitchell T Allen B Cook, G Buckberry. Indoor location using trilateration characteristics. 2009.

[Bea09] Stéphane Beauregard. Infrastructureless pedestrian positioning. 2009.

[Bek12] Anja Bekkelien. Bluetooth indoor positioning. Masterthesis, University of Geneva, 2012.

[BLGSW12] J.A. Bitsch Link, F. Gerdsmeier, P. Smith, and K. Wehrle. Indoor navigation on wheels (and on foot) using smartphones. In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–10, 2012.

[Bol08] Philipp Bolliger. Redpin - adaptive, zero-configuration indoor localization through user collaboration. *MELT*, September 2008.

[BP00] P. Bahl and V.N. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784 vol.2, 2000.

[BWH+04] Simon Brugger, Michael Wahl, Severin Heil, Philipp Hallauer, and Corina Schiesser. Bedeutungen des bahnhofs bern aus der sicht der benutzer und benutzerinnen. *Universitt Bern Geographisches Institut*, 2004.

[Cha09] S.S. Chawathe. Low-latency indoor localization using bluetooth beacons. In *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pages 1–7, 2009.

[CL11] M. Constantinides S. Nicolaou D. Zeinalipour-Yazti C.G. Panayiotou C. Laoudias, G. Constantinou. A platform for the evaluation of fingerprint positioning algorithms on android smartphone. Poster, Sept. 2011.

[DANS+10] Alessio De Angelis, John Nilsson, Isaac Skog, Peter Händel, and Paolo Carbone. Indoor positioning by ultrawide band radio aided inertial navigation. 2010.

[Eck14] Christoph Eck. Navigation algorithms with applications to unmanned helicopters. `http://www.uav.ethz.ch/research/projects/navigation_algorithms/schematic_strapdown`, Jan 2014.

[Gri14] Steve Griffin. Calculating the altitude of a known pressure. `https://www.brisbanehotairballooning.com.au/faqs/education/113-pressure-altitude-conversion.html`, Feb 2014.

[GSDM11] B. Gozick, K.P. Subbu, R. Dantu, and T. Maeshiro. Magnetic maps for indoor navigation. *Instrumentation and Measurement, IEEE Transactions on*, 60(12):3883–3891, 2011.

[GSHD08] U Großman, M Schauch, S Hakobyan, and M Dalhaus. Rssi-basierte wlan-ortung mit pda (personal digital assistant) innerhalb von gebäuden. *Wireless Communication and Information: New Technologies and Applications*, pages 223–238, 2008.

[HÖ8] DI Udo Häberlin. (un-)sicherheit und urbanitÄt. *MA 18, Stadtentwicklung und Stadtplanung*, 2008.

[Hav12] Janne Haverinen. Ambient magnetic field-based indoor location technology. Technical report, IndoorAtlas Ltd., 2012.

[HKY09] Chulmin Jun Hyeyoung Kim and Hyunjin Yi. A sdbms-based 2d-3d hybrid model for indoor routing. In *Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*. IEEE Computer Society, 2009.

[HZH+12] F. Hoflinger, Rui Zhang, J. Hoppe, A. Bannoura, L.M. Reindl, J. Wendeberg, M. Buhrer, and C. Schindelhauer. Acoustic self-calibrating system for indoor smartphone tracking (assist). In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–9, 2012.

[IDV13] Andreas Walther Yue M. Lu Ivan Dokmani, Reza Parhizkar and Martin Vetterli. Acoustic echoes reveal room shape. *Physical Sciences - Applied Mathematics:*, 2013.

[IK12] Huikyung Oh Incheol Kim, Eunmi Choi. Indoor user tracking with particle filter. In *The Fourth International Conference on Advanced Cognitive Technologies and Applications*. IARA, 2012.

[Inc14a] Google Inc. Motion sensors. `http://developer.android.com/guide/topics/sensors/sensors_motion.html`, Jan 2014.

[Inc14b] Google Inc. Using the rotation vector sensor. `http://developer.android.com/guide/topics/sensors/sensors_motion.html`, Feb 2014.

[ind14] indoo.rs. indoo.rs. `http://indoo.rs`, Feb 2014.

[Jan07] Andreas Janzen. Position determination with wlan and bluetooth. Universitaet Freiburg, Institut fuer Informatik, 2007.

[Kal60] R. E. Kalman. A new approach to linear filfilter and prediction problems. *ASMEJournal of Basic Engineering*, D:34–45, 1960.

[KKYK12] Seong-Eun Kim, Yong Kim, Jihyun Yoon, and Eung Sun Kim. Indoor positioning system using geomagnetic anomalies for smartphones. In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–5. IEEE, 2012.

[KPR10] P. Kemppi, J. Pajunen, and T. Rautiainen. Use of artificial magnetic anomalies in indoor pedestrian navigation. In *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, pages 1–5, 2010.

[KW11] Moritz Kessel and Martin Werner. Smartpos: Accurate and precise indoor positioning on mobile phones. In *MOBILITY 2011, The First International Conference on Mobile Services, Resources, and Users*, pages 158–163, 2011.

[LCC+12] C. Laoudias, G. Constantinou, M. Constantinides, S. Nicolaou, D. Zeinalipour-Yazti, and C.G. Panayiotou. The airplace indoor positioning platform for android smartphones. In *Mobile Data Management (MDM), 2012 IEEE 13th International Conference on*, pages 312–315, 2012.

[LGDR12] Binghao Li, T. Gallagher, A.G. Dempster, and C. Rizos. How feasible is the use of magnetic field alone for indoor positioning? In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1–9, 2012.

[LGT12] Etienne Le Grand and Sebastian Thrun. 3-axis magnetic field mapping and fusion for indoor localization. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, pages 358–364. IEEE, 2012.

[LJYH13] Jun-Sung Lim, Woo-Hyuk Jang, Gi-Wan Yoon, and Dong-Soo Han. Radio map update automation for wifi positioning systems. *Communications Letters, IEEE*, 17(4):693–696, 2013.

[LKP09] C. Laoudias, P. Kemppi, and C.G. Panayiotou. Localization using radial basis function networks and signal strength fingerprints in wlan. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–6, 2009.

[LMP11] C. Laoudias, M.P. Michaelides, and C. Panayiotou. Fault tolerant fingerprint-based positioning. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5, 2011.

[Man08] Nico Manske. *Verfahren zur Positionsbestimmung im Indoor-Bereich.* PhD thesis, Fakultt Technik und Informatik Department Informatik, 2008.

[MCRG+10] Rob Methorst, Christine Chaloupka-Risser, Jrgen Gerlach, Shalom Hakkert, Clemens Kaufmann, Kimmo Lapintie, Lucia Martincigh, Hector Monterde i Bort, Diego Moreno Ribas, Nicole Muhlrad, Ralf Risser, and Thrse Steenberghen. Pqn final report - part b5 documentation. Online, November 2010.

[Meu11] Richard Meusers. Navigation in gebäuden: Google weist den weg zu kasse und klo. Online, 11 2011.

[MF08] Andreas Wüster Martin Forster, Edith Huber. Subjektives sicherheitsgefühl und Überwachung. In Donau Universitaet Krems, editor, *Information Security Conference*, pages 44–51. Donau Universitaet Krems, Donau Universitaet Krems, October 2008.

[Moe13] Carsten Moeller. Osm2po. `http://osm2po.de/`, May 2013. Version 4.7.7.

[MS08] Alexandra Millonig and Katja Schechtner. Understanding walking behaviour–pedestrian motion patterns and preferences in shopping environments. In *Walk21-9th International Conference on Walking*, 2008.

[MW13] Norbert Kühtreiber Manfred Wieser. Lecture notes integrated navigation. Paper, 2013.

[Ope14] OpenStreetMap. Indoorosm. `http://wiki.openstreetmap.org/wiki/IndoorOSM`, Jan 2014.

[P.07] Suyash P. Adaptive nonparametric markov models and information-theoretic methods for image restoration and segmentation. Online, 02 2007.

[PB09] Maurice Chu Marc Langheinrich Philipp Bolliger, Kurt Partridge. Improving location fingerprinting through motion detection and asynchronous interval labeling. *ETH Zurich*, 2009.

[PFS+12] Michael Peter, Dieter Fritsch, Bernhardt Schäfer, Alfred Kleusberg, Jó Agila Bitsch Link, and Klaus Wehrle. Versatile geo-referenced maps for indoor navigation of pedestrians. In *International Conference on Indoor Positioning and Indoor Navigation*, volume 13, page 15th, 2012.

[Reh07] Sarah Rehm. Emotionale beeinflussung des menschlichen navigationsverhaltens. Master's thesis, Fakultät für Biologie Eberhard-Karls-Universität Tübingen, 2007.

[Rie12] Julia Riegler. Sicherheit im öffentlichen raum. *Öffentliche Sicherheit*, 11-12:40–41, 2012.

[Röh09] Christof Röhrig. Lokalisierungsverfahren für drahtlose sensornetzwerke. *Herzog MH (Hrsg.): Wireless Communication and Information–Radio Engineering and Multimedia Applications. Boizenburg, Deutschland: Verlag Werner Hülsbusch*, pages 81–97, 2009.

[RPJ97] G Rengert and W Pelfrey Jr. Cognitive mapping of the city center: Comparative perceptions of dangerous places. *Crime mapping and crime prevention*, pages 193–218, 1997.

[RSA+13] Mirco Rossi, Julia Seiter, Oliver Amft, Seraina Buchmeier, and Gerhard Tröster. Roomsense: an indoor positioning system for smartphones using active sound probing. In *Proceedings of the 4th Augmented Human International Conference*, pages 89–95. ACM, 2013.

[RVA07] Volkan Cevher James H. McClellan Rajbabu Velmurugan, Shyam Subramanian and David V. Anderson. Mixed-mode implementation of particle filters. 2007.

[RW12] Judy Kay Bob Kummerfeld Michael Fry-Tsvi Kuflik Rainer Wasinger, Kurt Gubi. Roughmaps - a generic platform to support symbolic map use in indoor environments. *International Conference on Indoor Positioning and Indoor Navigation*, 2012.

[SALP09] Ville HONKAVIRTA Simo ALI-LYTTY, Tommi PERL and Robert PICH. Fingerprint kalman filter in indoor positioning applications. *3rd IEEE Multiconference on Systems and Control*, pages 1678–1683, 2009.

[Sch13] Patrick Schnabel. Ortung und positionsbestimmung mit mobilfunk. `http://www.elektronik-kompendium.de/sites/kom/1201061.htm`, Dec 2013.

[SCM10] Alberto Serra, Davide Carboni, and Valentina Marotto. Indoor pedestrian navigation system using a modern smartphone. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, MobileHCI '10, pages 397–398, New York, NY, USA, 2010. ACM.

[SR11] Ubejd Shala and Angel Rodriguez. Indoor positioning using sensor-fusion in android devices, 2011.

[SSO08] Edgar-Philipp Stoffel, Korbinian Schoder, and Hans Jürgen Ohlbach. Applying hierarchical graphs to pedestrian indoor navigation. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, GIS '08, pages 54:1–54:4, New York, NY, USA, 2008. ACM.

[TFBD01] Sebastian Thrun, Dieter Fox, Wolfram Burgard, and Frank Dellaert. Monte carlo localization: Efcient position estimation for mobile robots. *Artificial Intelligence*, 128(12):99 – 141, 2001.

[Wen89] Lin Wenzhao. Magnetismus und der kompa. In *Wissenschaft und Technik im alten China*, pages 146–158. Birkhuser Basel, 1989.

[XPD09] J Xia, D Packer, and C Dong. Individual differences and tourist wayfinding behaviours. In *Proceedings of 18th World IMACS Congress and MODSIM09 International Congress on Modelling and Simulation, Cairns, Australia*, pages 1272–1278, 2009.

[You14] Gary Young. The fundamentals of in-building wireless solutions. https://www.bicsi.org/pdf/presentations/northcentral10/The%20Fundamentals%20of%20In-Building%20Wireless%20Solutions%20-%20ADC.pdf, Jan 2014.

[YZP13] Zhou Yong, Cai Zehui, and Chen Pengpeng. Indoor pedestrian positioning tracking algorithm with sparse anchor nodes. *International Journal of Distributed Sensor Networks*, 2013, 2013.

[Zho05] Rui Zhou. *Architecture and implementation of indoor wireless positioning system*. PhD thesis, Masters thesis, Department of Computer Science, Faculty of Engineering, the University of Freiburg, Germany, 2005.

[ZM11] Haiyang Zhang and F. Martin. Robotic mapping assisted by local magnetic field anomalies. In *Technologies for Practical Robot Applications (TePRA), 2011 IEEE Conference on*, pages 25–30, 2011.