Geographical Information Management and Applications

GIMA

Planning an indoor navigation service for a smartphone with Wi-Fi fingerprinting localization

Author: Justin Stook

E-mail: justinstook@gmail.com

Supervisor: Ir. Edward Verbree (TUD)

Professor: Prof. dr. ir. Peter van Oosterom (TUD)

Reviewer: Dr. Corné van Elzakker (UT-ITC)

Version (Final): 1.6 – August 18th , 2011

Universiteit Utrecht TUDelft ITC Wageningen UR

Dedicated to my father – may he navigate well in an unknown world to us

Selamat jalan

“The wisest men follow their own direction” – Euripides (484 BC – 406 BC)

Prologue

Navigation has become booming business recently. Well-known navigation services, such as TomTom, Garmin and Google Navigation, are dominating the market for outdoor, road based navigation. The fact that such applications are able to determine a location on a platform has always fascinated the author. Simple location determination is further utilized in specialized applications and functions, such as SportyPal, which logs the movements and interpolates several units, or the geotagging of photos. And so, this further piqued the interest to investigate matters in the indoor environment, as *one has to look beyond what one sees*.

However, navigation as a word can also be used as a metaphor to describe one man's life journey. How can one navigate as a human being? On the paths of life, one has to navigate through a web of complicated matters by fulfilling expectations and by making decisions that will influence your future. Decisions might be based on past experience and/or knowledge, there is no 'turn back' option, and things might heavily alter the path of life in unforeseen circumstances, both positive and negative. Navigation is then based on the abstract level.

The author, too, almost navigated towards another direction. Unforeseen circumstances in the private life forced him to choose a different flow than intended. The result is this thesis project, in which much time and effort has been put. The author hopes that the thesis is readable and understandable. He hopes that *every wise man should follow his own direction* and that as such, navigation applications will contribute in making those decisions.

Justin Stook

Acknowledgements

I would like to thank the following persons who helped me with my thesis research:

- Geert Wirken for his advice and excellent help op Java programming – we have spent a lot of time together tackling the problems!
- Mehmet Yildirim from the TU Berlin, for providing me excellent help as well, with the location determination scripts, as well as helping me out to get familiar with the Android environment.
- Gavin Brown from the TU Berlin, for providing me new insights of this domain.
- My supervisors Edward Verbree and Peter van Oosterom for their endless patience and who always believed I could do it, and provided me extensively feedback I could use.
- Joel Haasnoot for his advice and help on Java programming.
- My family and friends who (mentally) supported me to accomplish this feature.

Table of Contents

List of figures and tables

Main thesis

Annexes

Summary

One possible way to determine a position in the indoor environment is by using (the reflection and absorption of) signal strengths, also called multipath, from Wi-Fi routers. By recording the signal strengths in dBm over time at a certain position, a fingerprint can be created. These fingerprints can be unique in the sense that it is possible to distinguish positions. However, multipath are both a bless and a burden: the set of signal strengths are made because of the reflection and absorption, but at the same time, they are prone to changes in time, since at a position, signals can be received directly and indirectly. In this sense, it is not possible to record unique fingerprints at each single spot consistently, since the differences would not be significant to distinguish.

In this thesis, research has been done on how to make use of fingerprints in a smartphone based indoor navigation application, which uses fingerprints as a positioning technique, and which does not uses geometric maps or coordinates as a guideline. Since the fingerprints are directly translated to relative locations, such as “Room 2.200”, the method of positioning has been changed to localization instead of positioning. The focus is on the confirmation of the appropriate location regarding the requested route in topological sense: a user traverses a route constructed by nodes and edges without any explicit coordinates. The framework of the Open Geospatial Consortium’s (OGC) Open Location Services (OpenLS) has been used a guideline to set up a prototype. Parameters cannot be used, since those involve maps and WGS84 coordinates, while the project aims at the non-geometric features.

Signal strengths from a set of Media Access Controls (MAC) which transmits the Wi-Fi signals, have been recorded at 40 locations at the OTB research centre in Delft. In the application a scan will be compared to the recorded signal strengths and if there is a match, the location will be returned. This location is being matched with the requested route and it is possible to tell whether the user is at the correct location or not. As with the nature of multipath, one has to take into account the matching signal has to be within a range of the recorded signal: if the recorded signal was -65 dBm, sufficient search space has to be created around this signal. In this project, a search space of +4 dBm and -4 dBm has been proved sufficient. This means that a live signal of -67 dBm would provide a matching location, since this is within range of the original recorded signal (-69 < RSSI < -61).

The strength of this methodology is that it is easy to maintain a database with the recorded signals, although the recording, or site surveying, is rather time consuming, and the MACs can be easily replaced or removed. Multipath itself is inherent to changes in the time domain, which results in delays upon live tracking. As such, the fingerprint database requires a frequent update. Another strength is that there is no need for additional data processing: since there are no maps needed, there is a fast processing, as a certain degree of map matching does not apply here.

From the results, it was visible that it is highly recommended to store only mac addresses with their signal strengths, which are in the same physical space as the location fingerprint, as it improved the returning results on location predictions. Moreover, upon surveying, it is recommended to use the same class of receiver as the application is to be installed. Scan results obtained from high-end receivers can be considered as incompatible with scan results obtained from a medium-end receiver such as a smartphone. It is most likely the survey results from the latter will match the live results. As a result, a disadvantage is that separate fingerprint databases needs to be recorded for specific devices.

Further improvements are suggested regarding the translation (geocoding) of the fingerprint locations to absolute positions based on coordinates (allowing graphs or maps to be displayed), using extensions of smart navigation (as now, neutral directions have been used) and by distinguishing certain user properties (such as navigating through restricted areas, navigation for disabled users, or time of arrival estimations). In the developed prototype, room has been reserved for extensions such as these. The database for MAC and locations can also be placed on a server, rather than in an inherent database included in the application, for up scaling towards a large coverage area.

Issues will remain regarding the privacy of MAC addresses, since it is not allowed to record MAC addresses along with location information in several countries, as it is possible to trace identifiable persons, which contradicts legalities.

List of abbreviations and acronyms

3G	Third generation mobile telecommunication
A-GPS	Assisted GPS (see GPS)
AOA	Angle of Arrival
AP	Access Point
API	Application Programming Interface
BSSID	Basic Service Set Identifier, *same as MAC in value*
Cell-ID	Cell Identification
dBm	Decibel miliwatt
DSSS	Direct Sequence Spread Spectrum
FHSS	Frequency Hopping Spread Spectrum
GPS	Global Positioning System
IEEE	Institute of Electrical and Electronics Engineers
IEEE 802.11	*See WLAN*
IR	Infrared
IRDA	Infrared Data Association
LAN	Local Area Network
LOS	Line of Sight
MAC	Medium/Media Access Control, *same as SSID in value*
MIMO	Multiple Input Multiple Output
OFDM	Orthogonal Frequency Division Multiplexing
OGC	Open Geospatial Consortium
OpenLS	Open Location Services
POI	Point Of Interest
RFID	Radio Frequency Identifier
RSS(I)	Received Signal Strength (Indication)
SQL	Structured Query Language
SS	Sum of Squares
SSID	Service Set Identifier
TOA	Time of Arrival
UMTS	Universal Mobile Telecommunications System
UWB	Ultra Wide Band
Wi-Fi	Wireless Fidelity
WLAN	Wireless LAN (see LAN)
XML	eXtendible Markup Language

Terminology

Positioning vs. localization and accuracy vs. precision

A distinction can be made between positioning and localization. Haenselmann (2005) uses the following definitions:

- *Positioning* is the determination of a location with absolute (global) coordinates, such as (35.629535, 139.879818).
- *Localization* is the determination of a location with relative coordinates, such as "Room 5.12", and depends on the scale as such, e.g. a large room could be subdivided in smaller particles.

In this research, these two words are often mixed in use. When they are used in the context of describing a sensor technique, both refer to the same style of location determination. However, in the event these terms are used as a noun themselves, they are referring to their respective trait. In this perspective, the use of accuracy and precision, in combination with these techniques has also a distinction, as defined by the BIPM (2008):

- "[*Accuracy* is the degree of] closeness of agreement between a measured quantity value and a true quantity value […]. "
- "[*Precision* is the degree of] closeness of agreement between indications or measured quantity values obtained by replicate measurements on the same or similar objects under specified conditions."

A high accuracy means that measurement points (in this case location points) are close to the aimed target area. A high precision means that the measurement points together are close to each other This is displayed in Figure 1, where the purple dots represents the allocation of the GPS receiver, and the red dot represents the actual position.

This research is mainly focused on the accuracy being used, rather than the precision of the measurements of the signals.

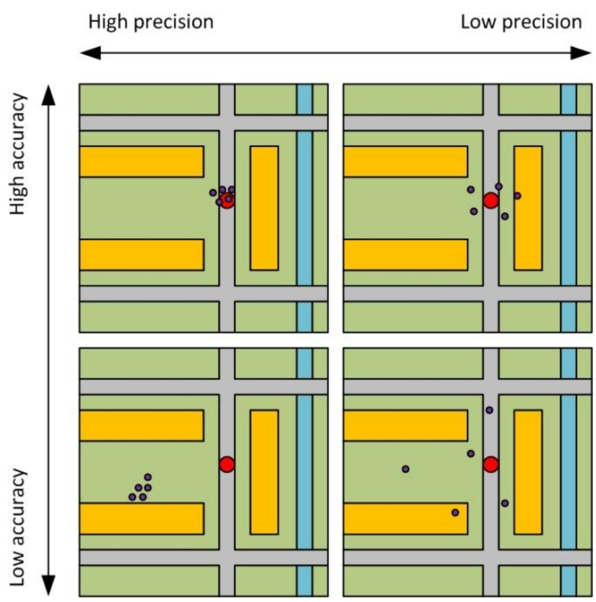


Figure 1. Accuracy and precision

Building geometry

In this thesis the combination of words *building geometry* is consistently being used. This can refer to:

- Creating geometry (building as constructing)
- The geometric features of a construction (the dimensions of a construction)

In this thesis, the second definition is being used. *Building geometry* refers to the dimension of the building as information and not to the development of geometric features itself.

PART 0 - Introduction

Contents

This thesis has been subdivided in five parts: part 0 with the general context of introduction, goal and research questions; part I with the main theory, part II with application oriented theory, part III with the application implementation and empirical research, and part IV in which the thesis will be concluded.

In this part 0, this report starts with a topic introduction and the structure of the thesis in chapter 1. Chapter 2 continues with a description of the goal, objectives and the scope (and limitations) of the research. Chapter 3 follows with the research questions.

1. Introduction

In this chapter the research topic (1.1) and the contents (1.2) of this thesis are described.

1.1. The research topic

Recent studies (Kolodziej & Hjelm, 2006; Zhang *et al.*, 2010; Dovis *et al.*, 2010) have shown the market in outdoor location based services (LBS) has increased over the past decade. With the use of a variety of sensor techniques, such as the Global Positioning System (GPS), BlueTooth, IEEE 802.11 Wireless Local Area Network (WLAN, also called Wi-Fi) and the Universal Mobile Telecommunications System (UMTS) and its generations in data transfer (e.g. 3G & 4G), software equipped mobile devices are capable to request and display vital on-the-fly information. GPS is the prime technology being used for these applications. Examples are the car navigation, in the form of TomTom or Garmin installed on independent devices and Google Navigation on cell phones, store locations requests (Albert Heins *Appie*), or friend finders (Google Latitude).

LBS allows on-the-fly navigation and service information, on-the-spot advertisement and notifications possibilities and other location related information, but are primarily outdoor, while the indoor possibilities have been neglected somehow (Manodham *et al.*, 2008, p. 296). One can think of navigating through a large stadium, a convention centre, an office complex or a railway station. But the application of indoor location information can also be used for robotics, for example. Indoor LBS especially proves its worth in emergency rescue or incident management, where high position accuracy is needed under difficult circumstances

However, GPS services have their limitations, which include a low accuracy in densely built environments and no available services at all inside buildings. Indeed, there has been, and still is, ongoing research about how these techniques can be improved (Kolodziej & Hjelm, 2006, Zhang *et al.*, 2010; Yim *et al.*, 2010; Dawes & Chin, 2010; Woo *et al.*, 2011, to name just a few). This research also focuses on the indoor navigation.

Despite the constraints of the current techniques and applications, existing sensor technologies and existing software can be enhanced to cater these difficulties. At the moment, Wi-Fi technology seems to be the most promising technique to detect indoor locations, although it has some weaknesses regarding its positioning accuracy. The question is whether it is possible to develop an indoor navigation system, which makes use of existing Wi-Fi technology. Furthermore, one has to deal with the target user (user profiles), to determine what support of applications and/or services is desirable, as this affects the general design of a service. As with sensor technologies, one must also take the objects and materials in account, as signal penetration behaves different for plywood than for concrete bunkers.

This research concerns about 1) the Wi-Fi fingerprinting technique and 2) the indoor navigation itself. From the latter, the focus is on how to use fingerprints to confirm where the user is, and whether the user is still on track in the context of the requested route towards the target, as this confirmation is part of navigation. Although various studies have dealt with Wi-Fi allocation processing as stated in Ibrahim & Ibrahim (2010), the novelty of this research that it will be strictly focused on the aspect of certainty and confirmation for civilian use.

The end product is a prototype for indoor navigation to be used on a smartphone, in particularly on an Android based device. The prototype can be used to navigate from one place in a building, to another place in a building, without having knowledge of the complex.

1.2. Contents of this thesis

First the objectives will be explained in chapter 2. Following these objectives, the research questions will be discussed in chapter 3. The content body of this thesis is divided into three parts.

In part I, the theoretical framework and background will be described in a full literature review. In chapter 4, some explanation about possible sensor techniques is provided. It also elaborates more on the principles of IEEE 802.11 (Wi-Fi) technology. Chapter 5 enhances this with positioning techniques, including fingerprinting. Chapter 6 deals with the navigation context, which forms another fundamental part of this research. Chapter 7 continues with the Open Location Standards (OpenLS) of the Open Geospatial Consortium (OGC), which can be used as a framework to develop a prototype.

Part II concerns with the application of the theoretical frame and is thus a mixture of both theory and practical issues and relies largely on a literature review, but allows room for ideas of the developing application. Chapter 8 deals with the application requirements, by dealing what is desirable: this is the part where the user preferences or user profiles will be stressed. Chapter 9 continues with connecting the sensor technique with an application, in which the OGC on OpenLS calls this gateway servicing. Chapter 10 deals with the implementation of navigational services, including algorithms and the use of OGC/OpenLS, in particular the directory and geocoding services. Chapter 11 follows with the modelling of the indoor environment. Chapter 12 proceeds with the output presentation of the chapters 8-11 and emphasizes on the visualization. Chapter 13 combines the previous chapters on the question how the navigation application can be constructed. Chapter 14 described how the architecture looks like.

Part III emphasizes on the results and provides detail on the prototype being exploited, which will be discussed in the chapters 15 (surveying methodology and results as well as database issues), 16 (localization methodologies) and 17 (layout and results).

Finally, part IV closes the thesis with the final conclusion in chapter 18. A reflection in the form of discussion on limitations and recommendations are to be found in chapter 19. In addition, remarks on privacy and legal issues form an epilogue in chapter 20.

Figure 2 displays how the chapters interrelate to each other.

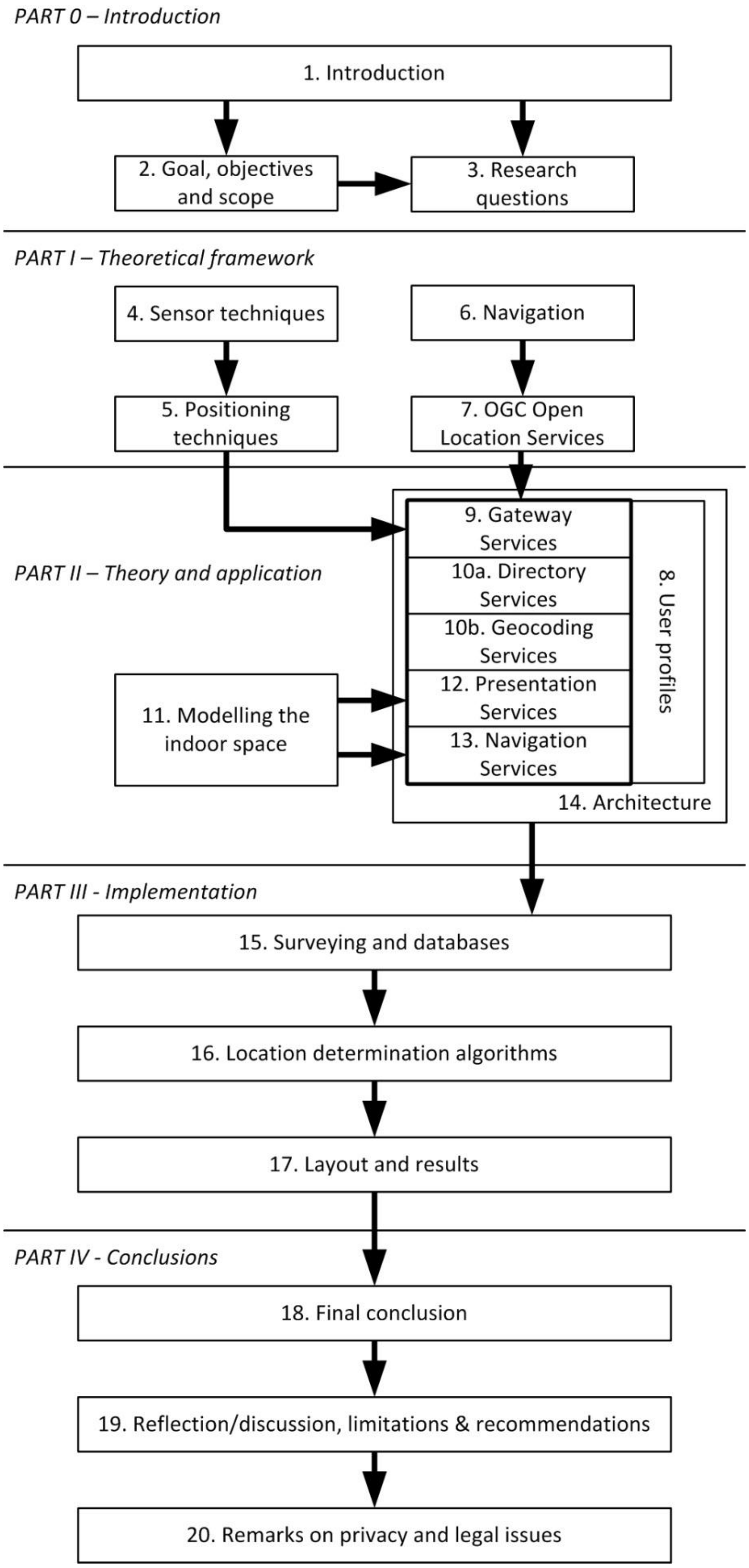


Figure 2. Interrelation of the chapters

2. Goal, objectives and scope/limitations

The goal is to investigate whether it is possible to use Wi-Fi fingerprinting technology as a positioning technique for indoor navigation, which relies on the framework of the Open Location Services standards, and which does not necessarily utilizes geometric features. The focus is on assessing the use of fingerprints as a confirmation tool, which will be incorporated in an application prototype.

The development of a prototype itself is subject to various factors which have to be taken into account, such as building material which will affect the Wi-Fi signal strengths and thus the positioning of the user. User profiles – described by ETSI (2010) as a “total set of user-related information, preferences, rules and settings” influences the development of a service as well.

Thus, the following objectives have been set for this project:

Regarding the theoretical context, it must become clear…

- Why Wi-Fi technology is being used, and why not other technologies.
- What infrastructural requirements are needed to develop this.
- What the target user and goal of the application should be.
- How it can be possible to build a system.

Regarding the application development, an application must be build, which…

- Its framework shall be based on the OGC Standards for Open Location Services (OpenLS).
- Suits the user profile and goal of the application (including the end visualization and style of interaction).
- Includes a method to actively utilize the route confirmation aspect of navigation.

Regarding the service deployment

- The application will have to be tested and demonstrated on a mobile device, to check if the positioning accuracy is acceptable.
- A site survey and positioning model will be carried out.

The project is carried out in the OTB building in Delft. In particularly, only the ground and first floor will be used, to determine if it is possible to navigate in the vertical dimension, and to make sure the prototype will work in a relative small area.

The data and software being used will be dependent on the target area. However, further research will be conducted about the database, server and spatial referencing system at a later stage. Yet, the proposed platform will be a HTC Desire Z Android 2.2.3 smartphone.

The scope of the project is limited to Wi-Fi localization technology and its application in an indoor location based service. In particular, the focus is on the navigation side, where the question is *how you can navigate from A to B if A and B are named locations inside a building?* This means that anything else is beyond the scope and will not be discussed, which includes other possible sensor techniques, economic viability and market deployment, privacy matters and an extensive positioning accuracy and positioning precision estimation. However, for the latter, an evaluation is proposed, to verify and demonstrate the usefulness of the created software. This means it is not the intention to modify existing accuracy and precision algorithms. Finally, inverse indoor positioning, as in placing Wi-Fi access points just outside the building to determine the position inside as described by de la Roche et al. (2010) has been acknowledged as a possible allocation technique, but is regardless beyond the score of this proposal as research in this field is relatively new and a significant uncertainty exist in various aspects. More about Wi-Fi functionality is described in chapter 5.

3. Research questions

From the previous chapter, the following main research question has been derived:

Is it possible to develop an indoor navigation service for a mobile platform, based on the use of only Wi-Fi fingerprinting technology and the framework of the Open Location Services standards, but without using building geometry?

From the main research question, the following sub research questions can be posed:

- *How can Wi-Fi be used in mobile indoor navigation?*
  - This research question addresses the motivation of the chosen localization technique/approach. There are multiple techniques possible, but upon answering this question, it will become clear why Wi-Fi technology has been chosen in favour of other techniques, and how Wi-Fi fingerprinting localization actually works.
  - Part of navigation is the regular check whether the user is still at the right place. This assumes a route is present. Route confirmation is a fundamental aspect in this sense. The question is how Wi-Fi Fingerprints can handle this.
- *How can an existing infrastructure be used and modified for indoor navigation?*
  - Creating a new infrastructure is in the first place considered extremely time consuming as the thesis research is constraint by time limits it is not desirable to create it. Therefore, minor modifications to the infrastructure are ought to be more realistic, e.g. readjusting router access points placements. In the end, it will become clear what infrastructure is needed, and how the different components are linked together.
  - This also includes the use of some sort of reference system. Where traditional outdoor absolute (*x,y,z* or *φ,λ,h*) reference systems are being used, the indoor environment requires a coordinated based scale of detail, hence the outdoor referencing system is considered unsuitable. However, a true reference system can be neglected if the emphasis is on the fingerprints as a confirmation tool. Since each position has a unique set of fingerprints, one only need to know if the receiver is at the right position as that has been told.
- *How can the creation of an application be possible, without using building geometry?*
  - This question features the inclusion of two problems that in general mobile users will face, and must be addressed in the creation of the application.
    - First, the limited accuracy of the position. Even when the position deviates significantly from the true position, the question is if this information still can be used, and how. How can an application be created without the inclusion of building geometrical features, but solely fingerprints?
    - Second, on a mobile device, smart designing is a key factor as well, since the device has limitations in data transfer and data visualization, due to device limitations such as screen size and processing time.
- *What are the benefits of using the framework of Open Location Service standards in an indoor navigation application?*
  - It is time consuming to implement an own set of standard procedures. Instead, existing standards for open location services from the Open Geospatial Consortium will be used upon designing the application, as it contains a large knowledge and expertise base.

However, the main question implies that a possible prototype will deviate from using the standards themselves. Therefore only the main aspects are probably a guideline upon developing a prototype. How does a framework of OpenLS fits into a prototype which does not follow the parameters themselves? How can these modified for indoor use? Which types are there? Are there already existing datasets and if yes, how are they being used.

- *How can Wi-Fi fingerprinting determine a location which does not utilize geometric features, and how can it be implemented in an application as such?*
  - A final question deals with how Wi-Fi fingerprinting can be used as a localization technique. Upon surveying, one ends up with a comprehensive database, but how exactly can that information being used to reconvert it to location information?
  - As there is no geometry, it is difficult to tell a user to turn left or right as such, since there is no orientation aspect. Yet, a route which has been calculated from a node-and-edge based network is still suitable for navigation, although the network does not contain any coordinates. This means that the importance and usage of a topological network plays a key role upon answering this research question.

PART I - Theoretical framework

Contents

In this part, the pure theoretical contents are described. Chapter four describes several sensor techniques which can be used. It elaborates on why Wi-Fi is being used, thus attempting to answer the first research question *how can Wi-Fi be used in mobile indoor navigation?*

Chapter five continues with the localization techniques, with fingerprinting being the central method. Next, in chapter six, the navigational issues are considered. General thoughts of navigational algorithms, as well as the necessary equipment and reference systems are included here, which leads to an answer of the second research question *how can an existing infrastructure be used and modified for indoor navigation?*

Finally, in chapter seven, the use of standards is discussed, in particular the standards as set by the Open Geospatial Consortium.

4. Sensor techniques

This chapter deals with the sensor techniques. Section 4.1 describes briefly why GPS is not suitable for the indoor environment. Section 4.2 continues with alternatives for GPS, while section 4.3 picks up Wi-Fi as a technique to be used in this thesis. Section 4.4 closes with a general overview of Wi-Fi.

4.1. Why GPS cannot be used

At the moment, Global Positioning System (GPS, or more general: Global Navigation Satellite System, GNSS) and the concept of cell identification in the more mobile telecommunication systems (such as GSM and 4G) are the most dominant modes of techniques being used to determine the location of a compatible mobile receiver in the outdoor world. With the use of those signals, services can be deployed, such as retrieving information where a receiver is located. One of those services is navigation.

GPS is suitable for the outdoor navigation, as the navigable spaces are wide enough. The accuracy positioning of GPS is between 6 and 12 meters 95% of the time. The problem is that the reception of the GPS signals is bad in densely build areas, including the indoor environment. GPS has been enhanced in accuracy, but the accuracy positioning is still insufficient indoors, as it cannot solve the multipath interference, which is the reflectance (scattering) of the signal via objects (Manodham *et al.*, 2008). This is caused by the requirement of the line-of-sight (LOS) of GPS in order to function. When a user is spending time indoors, the GPS signals will not be able to penetrate through obstacles. Depending on the material and volume of the object, signals can be completely absorbed, reflected, or even when it does penetrate the obstacle, the signal possible come out weaker, (Swangmuang & Krishnamurthy, 2008), since GPS requires line-of-sight. Another issue that has been mentioned is the high power demand of GPS receivers, and that GPS often behaves unpredictable regarding its propagation conditions, with the result that the true location of the user deviates significantly than desired (Dovis *et al.*, 2010). Finally, GPS performs poorly in the vertical position. As stated above, accuracy between 6 and 12 metres is mostly reached for the outdoor environment, which is sufficient for its location based services. Indoor, that same accuracy is not wishful, as 1-2 metres is needed, to prevent the user to be allocated on the wrong floor, or in the wrong room. The 3D environment is thus extremely important for indoor location based services.

If a closer look can be taken on the outdoor navigation services, one can see some discrepancies, too. The user usually finds itself at some distance of the modelled environment. It might be possible the entire road network is modelled as a large network of nodes and edges with geometry. The navigation software allocates the user's location somewhere on the network. The accuracy is important to provide the user with a list of directions. An accuracy of 10 metres would be sufficient in the context of car navigation. The allocation of the user's position on the network meets the requirement, as displayed in the left half of Figure 3. The GPS of the user will not be allocated improperly on the road to the south. However in a maze with high walls, and narrow paths, a higher accuracy than 6 meters is needed, in order to navigate properly. If the sensor and the service do not meet the desired accuracy, both might not be usable at all, as displayed in the right half of Figure 3, where the user will be prompted to turn right, while there is no turn at all. One could compare this simulation with an indoor environment. The aim is to ensure at least an accuracy of 2 metres to navigate the user.

GPS is thus unsuitable for the indoor environment. In the next chapter, other sensor techniques are being described, with the emphasis on Wi-Fi technology.

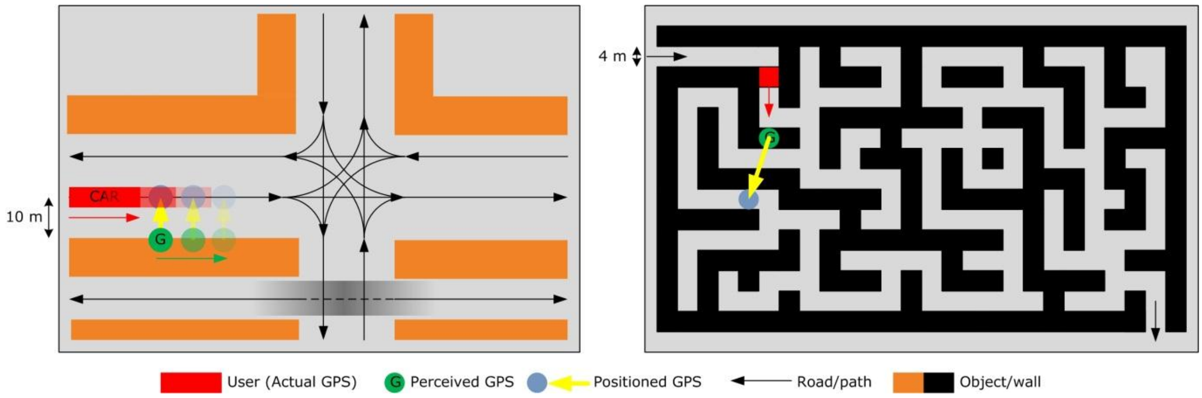


Figure 3. GPS positioning on the road and in a maze to simulate indoor navigation

4.2. Other sensor techniques compared

Indoor signal measurement is difficult, since the majority of the signals hardly (or cannot) penetrate through objects, but are being reflected instead. Other factors that decrease the reliability and accuracy includes the hardware of the transmitter, construction objects such as walls and doors, including their material, variations in temperature and humidity that might disturb signal strength, access constraints due to opening hours or security zones, and human activity (Mehmood *et al.*, 2010, p. 1220; Lim *et al.*, 2010, p. 406).

Kolodziej & Hjelm (2006), Manodham *et al.* (2008), Lim *et al.* (2010) and Woo *et al.* (2011) outlines a variety of sensor technologies that might be used for localization. Amongst these, Infrared, Ultrasound, Radio Frequency Identification (RFID), IEEE 802.11 (Wi-Fi) and Ultra Wide Band (UWB) are commonly named techniques. Infrared (IR; and the extended Infrared Data Association, IRDA) technology allows easy positioning due to ID matching of devices, but due to its short range and line-of-sight requirement, it is not a suitable system for localization. Radio Frequency Identification (RFID) makes use of tags and readers, which allow fast read/write data communication, but there is no real localization and positioning involved. Ultrasound technology, based on emitting sound pulses, is using the response time to determine a location, but it cannot countermeasure the problem of multipath effects and its weakness to environmental noise. Bluetooth is forwarded as a solution, since it is capable of high speed data transfer, within an acceptable range, but since the entire network is portable, positioning can be difficult. Ultra Wide Band (UWB) offers fast, low-power and large capacity data transfer, and has similar traits as both Bluetooth as Wi-Fi. The advantage is that it is capable of very precise localization (near 15 cm) but it is rather expensive. Finally, The IEEE 802.11 (Wi-Fi) technology of data exchange proves to be the most balanced, as the signal range of around 30-40 meter indoor is acceptable and its economic viability is stable. The technology is still being developed, with the current state of 802.11n reaching up to an indoor range of about 91 m, and an outdoor range of 182m (Ibrahim & Ibrahim, 2010). However, the localization accuracy at its current state is not sufficient and today there is still research to be conduct regarding which filters should be applied. Another weakness is that in larger buildings a large amount of transmitters needs to be installed, although the same applies to the other techniques.

Despite of each technology's shortcomings, Wi-Fi is considered the most attractive alternative to GPS for indoor usage, as its range is at a fair level (≈ 32 m), and Wi-Fi is widely available in many places, which saves significantly time and effort in arranging an infrastructure. Nevertheless, it is also possible to combine the different technologies in one application. For instance, an application that is capable of toggling between GPS, Bluetooth and Wi-Fi at will. However, it is not the goal of this thesis doing so, although it might serve as a recommendation for future work.

A brief summary of the technologies and techniques is shown in *Table 1*. A full overview is found in the annex, *Table AP. 1*. The allocation techniques, also known as the positioning techniques, are further described in chapter 5.

Technique	Range / accuracy	Remarks	Allocation meth.*
(A)GPS	Accuracy: 6.0 m - 10.0 m	+ Low barrier entry - Slow computation and processing time - Very susceptible to reflectance and multi-paths	Trilateration
GSM / UMTS	Range: ≈ 35.0 km	+ Globally available - Cell-based accuracy	Cell-ID Signal Strength
IR	Range: 0.7 m – 2.5 m	- Short range of detection limits infrastructure - No penetration of materials / multipath - Line of sight - Signal can be disturbed easily	Cell-ID
IRDA	Range: ≈ 2.5 m	*Same method as IR, yet with higher speed*	Cell-ID
RFID ***Active*** ***Passive***	 Range: ≈ 100 m Range: 1.5 m – 2.0 m	+ High-speed response time + Read/write capabilities - No communication network - No allocation information	Cell-ID
Ultrasound	Accuracy: 3.0 cm – 1.0 m	- No penetration of materials / multipath - Extremely sensitive to environment	Trilateration
Bluetooth	Range: ≈ 100 m Accuracy: 10 m – 20 m	+ High speed data transfer - Positioning via triangulation (no objects into account) - Explicit links between devices required - Its mobility also limits positioning and topology	Trilateration Signal Strength
Ultra Wide Band (UWB)	Range: ≈ 50 m (cap) Accuracy: 15 cm – 4 m	+ Multipath immunity/high precision) + High speed data (nearly 10xWi-Fi) - Not everywhere legal - Economically expensive	Trilateration Signal Strength
IEEE 802.11 (Wi-Fi)	Range: ≈ 32 m (indoor) ≈ 95 m (outdoor) Accuracy: 1 m – 5 m	+ Large scale available over the world + Economical viable - High power consumption - Slightly multipath susceptible	Trilateration Signal Strength

Table 1. Overview selection of possible techniques for localization

Table based on Manodham *et al.* (2008) and Kolodziej & Hjelm (2006) and Lim *et al.* (2010). *See also* Table AP. 1 *for a full overview*

**The allocation (positioning) methods are further explained in chapter 5.*

4.3. General background of Wi-Fi

Wi-Fi is the common name for the standard framework of a Wireless Local Area Network (WLAN), as set up by the 802 committee of the Institute of Electrical and Electronics Engineers. In fact, the official name is IEEE 802.11. The commonly standards that are being used are the 802.11b, 802.11g and the 802.11n versions. The contents below are based on the IEEE 802.11-2007 document, which can be retrieved at their website

http://standards.ieee.org/about/get/802/802.11.html.

4.3.1. Use of spread spectrum

IEEE 802.11 technology makes primarily use of modulation techniques in the 2.4 GHz industrial, scientific and medical radio band (ISM), or in the lesser used 5.0 GHz band. The radio waves make use of the spread spectrum modulation technique, which means that the generated carrier waves are over the full frequency domain, thus wider than the information signals, as displayed in Figure 4. There is a transmitter, usually a router, and a receiver, usually a computer or mobile device. The modulation technique for the 802.11 depends on the version being used, as displayed in Table 2.

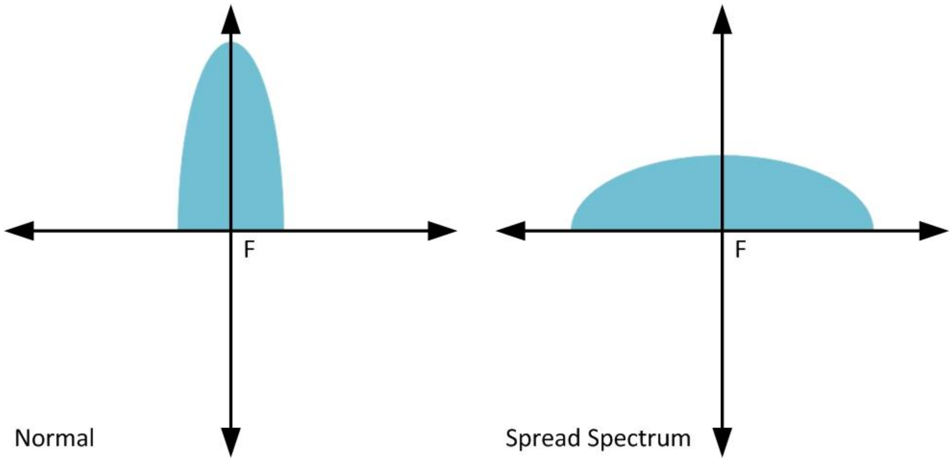


Figure 4. Normal versus Spread Spectrum

Image derived from Kessler (2006)

802.11	Frequency (GHz)	Bandwidth (MHz)	Data rate per stream (Mbit/s)	Allowable MIMO streams	Modulation technique
	2.4	20	1, 2	1	DSSS, FHSS
a	3.7, 5.0	20	6, 9, 12, 18, 24, 36, 48, 54	1	OFDM
b	2.4	20	5.5, 11	1	DSSS
g	2.4	20	6, 9, 12, 18, 24, 36, 48, 54	1	OFDM, DSSS
n	2.4, 5.0	20 40	7.2, 14.4, 21.7, 28.9, 43.3, 57.8, 72.2 15, 30, 45, 60, 90, 120, 135, 150	4 4	OFDM

Table 2. Properties of IEEE802.11 protocols

Source: IEEE (2011) and Wi-Fi Alliance (2009)

The original 802.11 and subsequent 802.11a are no longer in use today. The original 802.11 made use of Infrared technology, the Frequency Hopping Spread Spectrum (FHSS) or the Direct Sequence Spread Spectrum (DSSS), but was rather slow due to its low data rate. The 802.11a has become obsolete as well, because the signals of the Orthogonal Frequency Division Multiplexing (OFDM) modulation technique at the frequency of 3.7 GHz and 5.0 GHz could not penetrate well through walls, despite their higher frequency and their higher data stream. The 802.11b and the 802.11g are nowadays commonly used protocols, again operating at the 2.4 GHz frequency, knowing that interference with appliances in the same frequency (such as microwaves and Bluetooth) still occurs. Using the DSSS at this frequency ensured a reliable working. The main differences between the b and the g protocols are the levels of data rates, and the modulation techniques of OFDM in addition (yet instead of the 3.7 or 5.0 GHz level, at the 2.4 GHz level). Finally, the 802.11n was based on prior protocols, and by allowing multiple input multiple-output antennas (MIMO) (Wi-Fi Alliance, 2009; IEEE, 2011). This means that more than one transmitters and receivers are being used to increase capacity.

The Frequency Hopping Spread Spectrum (FHSS) is the transmitting of radio wave signals that constantly change between the frequencies. For the originally IEEE 802.11, this was less suitable, even though the security is rather high, because it is hard to intercept these.

Both the 802.11b and 802.11g make use of the DSSS modulation technique, with 802.11g also capable of using the OFDM technique. The 802.11n explicitly uses the OFDM, also at the 2.4 GHz.

In DSSS, waves are being send, a pseudo noise (PN) code symbol (a chip) is being transmitted, which is shorter and faster than an information bit. The data transmissions are being multiplied with a noise transmission, which is a sequence of 1 and -1 values, at a frequency higher than the original signal. A receiver can use the same PN sequence to reconstruct the data signals. This is displayed in Figure 5.

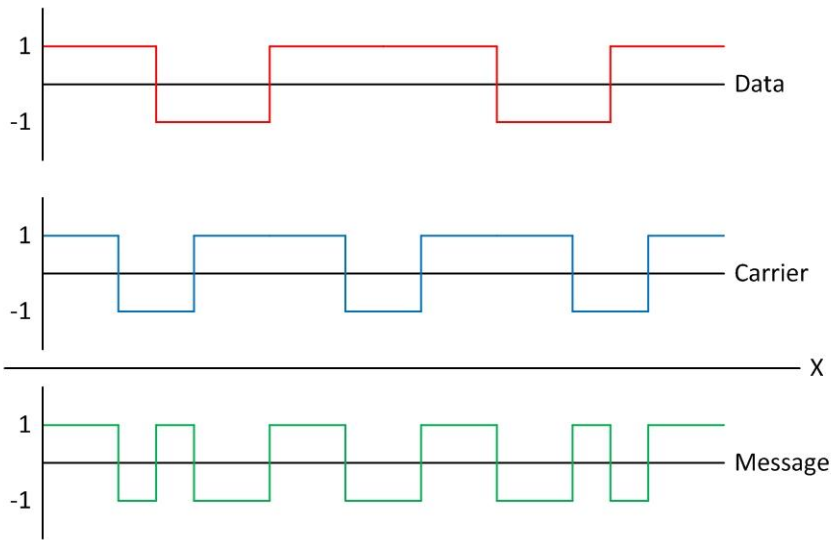


Figure 5. DSSS messaging

Image derived from Kessler (2006)

OFDM for 802.11 divides the 2.4 GHz band in 14 channels, each with a width of about 20-22 MHz wide, and 5 MHz apart. Not all channels are available (legal) in each country, such as the 14th channel, which is only available in Japan. In Figure 6, an example is provided. Within that channel, conventional data is being sent. The advantage of this technique is that it can switch to other channels in case problems are detected that interferes with the signal, such as Bluetooth or microwaves due signal loss despite being closely located to the access point. Most importantly it is possible to use overlapping channels for signal transmittance, countering selective fading unlike the case being for a single channel (Intini, 2000).

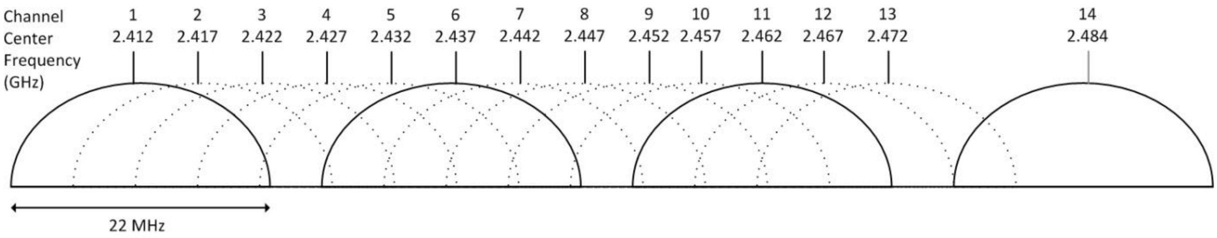


Figure 6. Channels of the 2.4 GHz for OFDM

Image derived from Intini (2000)

4.3.2. Frames / layers

The signals that 802.11 broadcast, contain frames, or also called layers. Each frame contains a Multiple Access Control (MAC) address. In combination with the received signal strength, expressed in dBm, it is possible to estimate how close an access point is. The IEEE 802.11 documents specifies in great detail what the wide array of the exact contents. The Wi-Fi Virtual Laboratory led by Thomas Sturgeon (2011) graphically explains briefly what a signal contains, as displayed in Figure 7; while Intini (2000) explicitly visualize the coding in Figure 8.

For Wi-Fi Fingerprinting, the only information that is required is the information in the management frames, since the signals are only being used to determine a location, and not to transfer data. Fingerprinting is described in the next chapter, among other techniques.

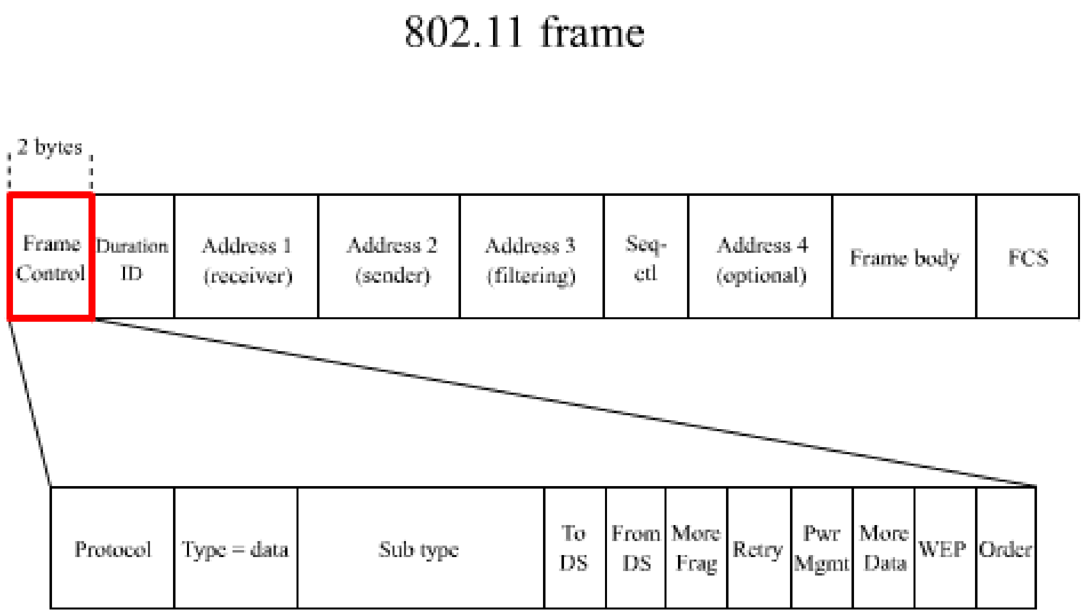


Figure 7. Frames of a 802.11 signal

Screenshot taken from the Wi-Fi Virtual Laboratory (Sturgeon et al., 2011)

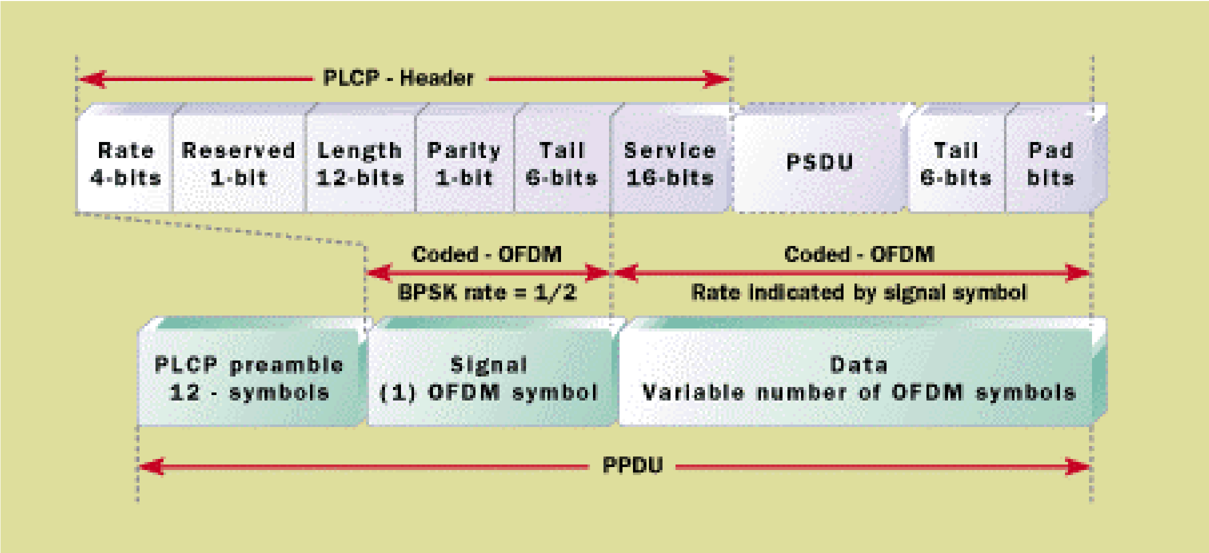


Figure 8. 802.11 Frames and its coding

Source: Intini (2000)

4.4. General strengths and weaknesses

Wi-Fi system is low-cost and low-entry: Wi-Fi access points are nowadays widely available. The main strength is that it is able to partly penetrate through objects. However, some weaknesses are the security (easy to intercept, unless further encrypted) and interference with other applications in the same frequency, such as Bluetooth and microwaves, although this depends on the channel being used in the case of OFDM. Limited data bandwidth is considered another limitation: Lim et al. (2010) for example noted the decrease of data transfer speed and bandwidth during office hours. In Table 3, a summation of the SWOT analysis can be found.

SWOT	Aspect	Remark
Strengths	1. Low-cost and low-entry	1. Widely available at affordable prices
	2. Able to penetrate walls in where GPS fails	2. Although effect is depleted after thick layers, such as thick concrete walls
	3. In available spaces, fairly good available signal strengths	3. Due to multi-path, good signal differentiation; up to 100m
	4. Specific location fingerprints available.	4. Coverage of entire building, if access points are well placed
Weaknesses	5. Susceptible to variations in signal strength due to time	5. A recorded fingerprint cannot exactly be reproduced
	6. Earthbound based, requires more infrastructure	6. For fingerprinting, more access points are needed, unlike satellite based
	7. Multi-path influenced by present objects	7. The more objects there are, the more differentiated the fingerprints will be
	8. Might interfere with other appliances in the 2.4 GHz ISM	8. For example Bluetooth and microwaves
	9. Site surveying and registering time consuming	9. Must be repeated for interior and movement changes, and for each building
	10. MAC address related – prone to changes	10. If system fails, or when access point fails, MAC cannot be used.
	11. Security is weak	11. Only applicable to data transfer
	12. Speed decreases with traffic	12. Only applicable to data transfer
Opportunities	13. Fingerprinting does not require geometric surveys	13. Time and effort can be saved on mapping
	14. Fingerprinting only necessary at selected places	14. It is not necessary to measure every *n* meter; only at places with important topological meaning or at least where fingerprints look different.
Threats	15. Bluetooth (BT) or Ultra Wideband (UWB) might overtake Wi-Fi for positioning	15. BT rather stable, UWB powerful in better catering multi-path + better range.
	16. Privacy related elements might block registration of MACs and thus fingerprinting	16. Issue of registration of private MACs – limitation to public buildings only

Table 3. SWOT analysis of Wi-Fi and Wi-Fi Fingerprinting

5. Positioning techniques

In the various literature studies, the word localization is being used to determine a position. However, in this research the term *allocation (method)* is being preferred, since it does not explicitly refer to an absolute or relative location, unless stated otherwise. See also the terminology at the beginning of this thesis research.

5.1. Five techniques

Different methodologies based on signal metrics can be used to allocate devices. Although there are more allocation techniques available, Woo *et al.* (2011), Zhang *et al.* (2010) and Kolodziej & Hjelm (2006) distinguished five main methods, using: A) Cell Identification, B) Angle of Arrival (triangulation), C) Time of Arrival (trilateration), D) Time Difference of Arrival and E) Received Signal Strength Categories, also called Fingerprinting.

A) With Cell Identification (Cell-ID), transmitters (or: access points) are dividing an area in tiles, or cells. Within those cells, the receiver can be detected. However, this method is imprecise, because it is hard to determine where in the cell the receiver can be spotted.

B) The Angle of Arrival (AOA) determines the position of the user by measuring the angle towards the receiver from the transmitter (Woo *et al.*, 2011). The transmitters must be capable of calculating such information. This can be done with directional antennae. Yet, the method is unreliable, since it is prone to multi-paths, plus it requires the line of sight to detect the receiver, which is hard to counter in the harsh indoor environment (Zhang *et al.*, 2010; Kolodziej & Hjelm, 2006). In the extension of the AOA, triangulation can be done to map an entire region, based on the angles.

C) Time of Arrival (TOA) measures the exact distance by using the travel time of the signal from the transmitter to the receiver. Using the equation *R = time x speed*, where *speed* is a constant, only *time* needs to be measured to determine the exact location *R* (Zhang *et al.*, 2010). However, in order to get more accurate results, synchronization of the receivers is needed.

D) TOA requires synchronization of both transmitters and receivers, whereas the Time Difference of Arrival (TDOA) requires synchronization of the receivers (Woo *et al.*, 2011). Both methods belong to the trilateration group, as it involves the intersection of the radii of the transmitters (Zhang *et al.*, 2010).

E) Lastly, the Received Signal Strength (RSS), also called fingerprinting, where a radio map is being created. There is an *offline* and an *online* phase, also called the *training* phase and *tracking* phase. The offline phase of fingerprinting is determining the signal characteristics of a given point and storing it in a database. The online phase is picking up the signal characteristics and compares it with the database, so that the place can be determined. Ideally, a filter is needed to increase the positional accuracy and estimations of the locations, even though the reflection and absorbance of the signals have been taken into account (Kolodziej & Hjelm, 2006; Woo *et al.*, 2010).

Figure 9 presents an overview of the localization techniques mentioned. Figure 10 presents a more comprehensive illustration of the offline phase of fingerprinting. In the online phase, the device attempts to match the signal characteristics with its position and is in this way able to determine where it is located. In the next sections, the emphasis is on fingerprinting as a technique.

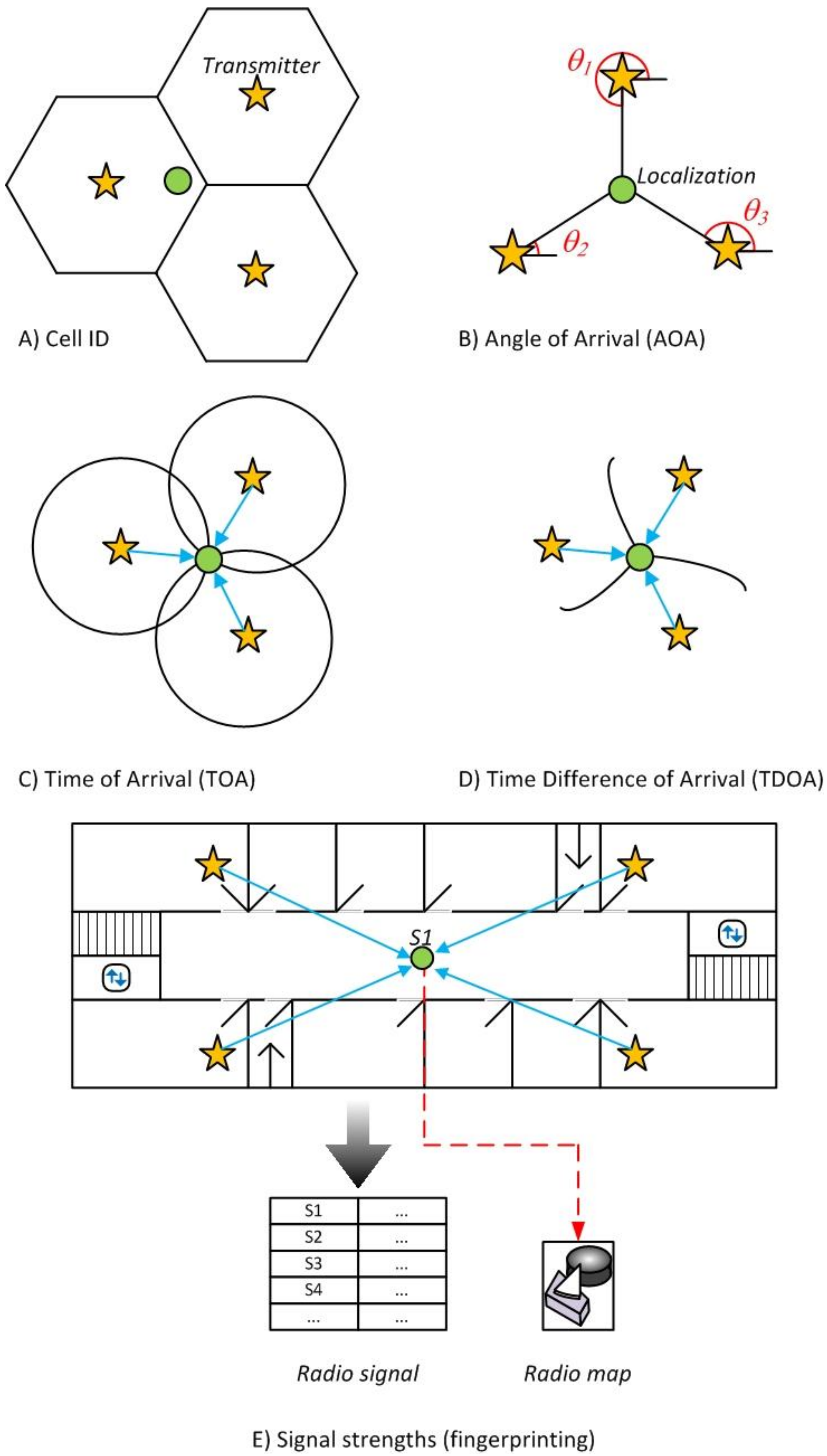
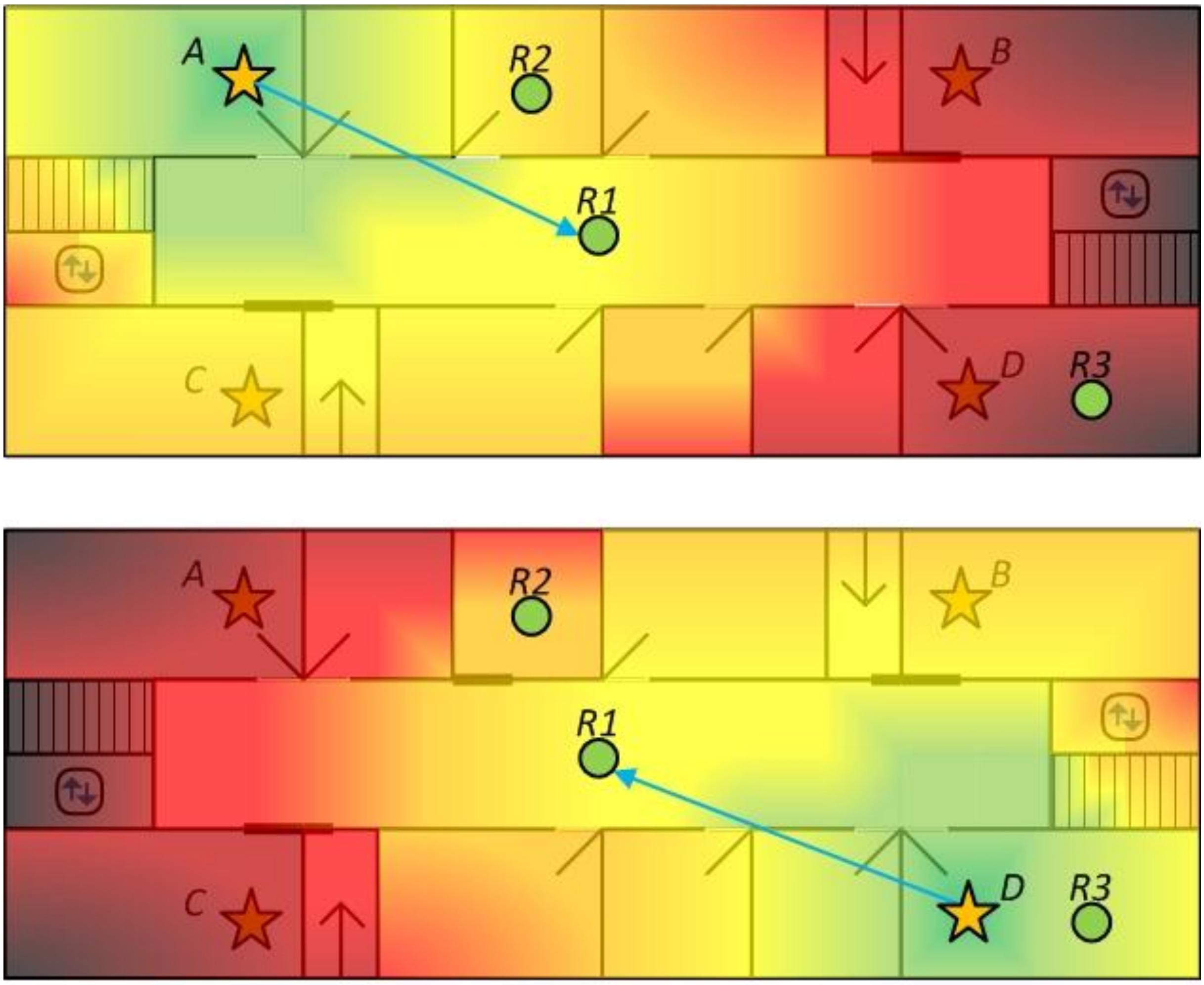
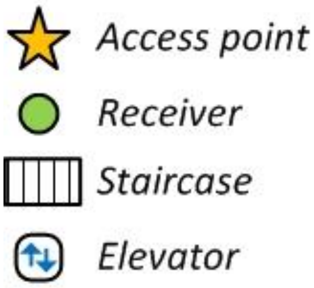


Figure 9. Allocation methods

Illustrations are based on Woo et al. (2011).





dBm	A	B	C	D
R1	-60	-65	-60	-65
R2	-65	-73	-75	-81
R3	-88	-60	-85	-45
...	...	...	...	...

Figure 10. An example of fingerprinting

Based on Woo et al. (2011). This is a pure theoretical illustration; in reality the signal strength distribution heavily depends on present objects, and overlapping coverage areas. In this figure, only the radio maps of access points A and D are displayed; in reality all radio maps together form one overview. As can be seen, this placement of Wi-Fi access points might not be the most efficient one.

5.2. Fingerprinting elaborated

Recent studies (Jan *et al.*, 2010; Lim *et al.*, 2010; Yim, 2008) have shown that fingerprinting is a fundamental method of indoor allocation, even though the level of accuracy is not at its desired level of approximately 2 metres, because of its relative robustness to handle multi-path effects. Still, various challenges exist. As soon as objects have moved (which includes the presence of peoples), the fingerprints might alter. In addition, the timestamps of the received signal strength (RSS) is not constant over the day. In public buildings for instance, the RSS is about 5 dBm less during office hours (from approximately 9 am to 6 pm), which could influence the fingerprint database, but not too much (Lim *et al.*, 2010), since the data layer of Wi-Fi is needed. In order to create a fingerprint database, one must be able to log the signal strengths throughout the entire building, with the help of data logging services (Dawes & Chin, 2011). That might cause problems in case of sealed off zones.

Various studies have introduced different filters for improving positioning on maps directly. Yim *et al.* (2010) used an extended Kalman filter to improve the accuracy of the Wi-Fi positioning during the offline fingerprinting phase. The Kalman filter basically reduces the noise in locational determination. However, the use of Kalman filters is disputed by various authors. Chiou et al. (2009) developed an alternative method for fingerprinting, which is called adaptive positioning but it still makes use of the Kalman filter. Jan *et al.* (2010) suggested that a particle filter would be better instead of a Kalman filter, since a Kalman filter is a linear model, suitable for linear environments. However, the real world is usually not a linear build environment. Jan *et al.* (2010) and Dawes & Chin (2011) however found out that Bayesian map matching algorithms have difficulties in environment with high variable transmitters and fingerprinting (RSS) values. They also showed that in the same fingerprinting phase, the accuracy is highest, when nearest neighbour filters are applied. The nearest neighbour assumes that a specific point that has to be measured is most likely similar, or at least close in value, to the most nearby points, depending on the weight (Roos et al., 2002).

An extensive offline/training phase will be carried out in the study area, and after that, the online/tracking phase will be tested for its accuracy. This will be taken into account for the evaluation of the application, as the user allocation will be used for the navigation. It is not the goal of the research to fully assess the accuracy of Wi-Fi fingerprinting, however. As a final remark, it is possible to determine signal strengths in the vertical position as well. If the router is attached to the ceiling, the received signal strength is a bit weaker near the floor. However, since it is unlikely for persons to keep their receiver at near 2 metres or at 0.1 metres above the floor, this research will only conduct a survey at the height of near 1 metre, which should be the height of the users to keep their phone in their hands. In addition, since it is not the aim of the study to enhance positioning accuracy (instead, localization is considered), a real map matching does not take place.

Due to high variations in signal strengths caused by multipath, the fingerprint database needs to be updated frequently, and it might be reproduced for a variety of devices, since a high-end Wi-Fi receiver perceives signal strengths to be stronger than low-end Wi-Fi receivers.

5.3. Building constructions influencing fingerprinting

As already pointed out, GPS suffers significantly from radio wave propagation (Jan et al., 2010). Signals are unable to penetrate buildings, and even if they do so, the signal strength significantly decreases. Although Wi-Fi belongs to the same category of frequency, the UHF (Ultra High Frequency, which is between 300MHz and 3 GHz), Wi-Fi is better able to cater the propagation as it is in the higher, longer range of the frequency: Wi-Fi at 2.4 GHz (although 802.11n is capable of using the 5.0 GHz band) and GPS L1 on 1.5-1.6 GHz and L2 on 1.2 GHz (Ibrahim & Ibrahim, 2010).

This does not mean it can perfectly handle this problem. Not only intensive use of the access point of Wi-Fi influences the signal strength, and thus fingerprinting, but so does the building constructions.

Klukas et al. (2004) studied the effects of building materials on the UHF signals and found evidence aluminium tissues and concrete, cinder block objects interferes the signal strength: a loss of 10-15 dBm for aluminium objects and even 20-26 dBm for cinder block objects. This is in contrast of the plywood and gyprock, where in both cases a power loss of between 0 and 2 dBm was found. A combination of the latter two materials resulted in moderate power loss of 5-10 dBm. This temporal aspect has been left out: in theory, an object can receive a signal via the line-of-sight (directly), but also via reflectance of other objects (indirectly). In this case, the receiver did receive a signal, but it was either delayed (indirect signal) or it was a weaker signal (direct signal).

In addition, Ibrahim & Ibrahim (2010) found out that the signal strength also sharply drops after a distance of 20m. This means that, regardless of the building construction and materials used, the placement of the Wi-Fi access points should be placed at a proper distance from each other, to ensure the fingerprinting results will not be heavily influenced (prone to alterations) by misplacements. Combined with the signal propagation as described by Klukas et al. (2004), great care has to be taken as well for the placement next to specific types of walls or ceilings as such. In Figure 10 for example, the walls of the elevator shaft might contain a thicker coating than normal, resulting in lower signal strength.

5.4. Using signal strengths to locate

The fingerprinting method itself is not new, as various (MSc) researchers have tried to use various filters upon location detection directly projected on maps, and at each possible location. Mostly, attempts have been made to calibrate the position in the indoor space. Lee (2007) addresses the paradigm of fingerprinting by converting discrete signal strength values to continuous values and implements various probabilistic filters to increase location determination for constant tracking at each single space. De Koning (2010), albeit with GSM signals outdoors instead of Wi-Fi signals indoors, first correlates signal space with Euclidean space and implemented variations on the weighted nearest neighbour filter, to retrieve location information. Yim et al. (2010) proposes a similar combination with weighted nearest neighbour filters and trilateration, but concludes that an extended Kalman filter would provide better results.

Milioris et al. (2010) proposed an analysis in which the distances in terms of signal strength of the training phase were compared with the tracking phase, and in addition, a multivariate Gaussian filter has been used to do so. Although attempts have been made to calculate mean errors, the idea of comparing them with each other itself is sound. Finally, Shum et al. (2011) have chosen a method to determine positions without filters, also by comparing differences in signal strengths in the training and tracking phases, but then apply a least sum of square methods, which has been regarded very promising.

In this thesis project and prototype, a similar approach to Shum et al. (2011) has been chosen, as with the lack of geometry, no direct map matching algorithms can be implemented, resulting in the fact that no filters, such as the Kalman or Weighted Nearest Neighbour, can be used. In addition, it is not the aim to pinpoint exact locations of the entire space, as it is assumed that locations which are very close to each other have similar fingerprints. This is why locations have been chosen and recorded, in which the fingerprints can be clearly distinguished from each other, as that was the assumption of Shum et al. (2011) similarly. However, the text format outputs being provided, can be geocoded into real world coordinates. To do so, two methods have been proposed: a count method, which only counts the number of matches, and a least sum of squares method as proposed by Shum et al. (2011). These methods have been chosen in the first place because there is no map matching which implies there is no need in exact positioning, and in the second place to underline the simplicity of the application. These two methods are described and compared in chapter 16.3.

6. Navigation

In this chapter general theory about navigation and its required infrastructure are described, which includes the general thoughts about navigation (6.1), as well as the importance of certainty and route confirmation (6.2), algorithms (6.3), required infrastructure (6.4), and spatial referencing (6.5).

6.1. What is navigation?

"Navigation is the process of monitoring and controlling the movement of a craft or vehicle from one place of another" – Bowditch.

This statement from Nathaniel Bowditch (2002) is still applicable. Navigation can also be described as going from source A to destination B, while knowing where you are from, what to expect on your way, while being in motion from source to destination (or differently said: while the source is moving towards a fixed destination). It assumes that a start and destination is known, and that one has to go from start to destination via a certain path. Present-day navigation is no longer solely based on the position of human interference with the eye line-of-sight such as in watch towers, celestial processes or dead reckoning. Today, navigation can be fully done without the use of any direct human interference, such as radio, radar and satellite navigation. As with these modern navigation techniques, *going* from A to B itself is no longer the main subject, but the question is rather *how* one should going from A to B and *how* one would retrieve information along the way. The aspects of monitoring and controlling is very important, with a rapid emerge of applications which utilizes navigation for information, ranging from tracking and tracing parcels, to chip-tagged products leaving unauthorized a shop (resulting in ringing alarms at the front gate of that same shop). In short, navigation requires information on:

- Where the user is (*start location*);
- Where the user would like to go (*destination*), which can be queried from an address, coordinate, points of interest, or a combination. This is strongly related to the use of user profiles to steer the end user; and
- How the user should go from start to destination: the information of the navigation itself, as well as information along the way, such as the availability of an ATM, depending on the user preferences and user profile. This is the path, and can be configured according to the wishes of the end user, such as the shortest or fastest path.

6.2. Navigation principles

6.2.1. General principles

In order to perform a navigation service, a start point, an end point, the navigable space (the network), (re)orientation and directives are required. All locational information between start and end point can be used for this desire, such as attributes of the start and end point which can be queried, fastest or shortest route (although pedestrian routes do not differ much regarding this aspect, unlike motorized transport, since the velocity of the walking user is usually constant), characteristics of the route in-between and its orientation. All of these sets need to be modelled somehow. Kolodziej & Hjelm (2006), mention the fundamental inclusions of the following procedures:

- Input module for start and destination of user;

- Route database with positions and connections (route network map);
    - Topology and structure is coupled to this route database; it is not absolutely necessary to have coordinates, however.
- Route calculation module, using the route database and algorithms such as Dijkstra's shortest path algorithms;
- Presentation of the route guidance (list of directives).

As soon as a route has been calculated, a list of directives and a map can be constructed. The route will be dependent on what the profile and wishes of the users are. For instance, the user is a guest in a large power plant, and cannot enter all zones due to security restrictions. Another example is disabled person, who can only take the elevator. This will clearly influence the route. Either way, the application has to keep track in which phase the user is currently at. Kolodziej & Hjelm (2006) approach this in such way, that each directive in the list has a start and an end; as soon as the end has reached, the directive will be removed from the list, and the next directive will start. This will continuously loop, until the destination has been reached.

6.2.2. Certainty and route confirmation

Indoor, the user has a wide array of recognition points available. Since the indoor environment is very detailed, the user easily passes his destination or important turns, the user cannot constantly watch his mobile screen for confirmation. The user wants to have a level of certainty.

In navigation, one fundamental technique is to check whether the user is still at the assigned route. The receiver will pass through certain points and if the position matches the predicted point, confirmation is being sent to the application the user is on route. If this is not being the case, a recalculation has to take place. In this thesis, the fingerprints can be considered as such check points.

6.3. Navigation algorithms

The navigation itself is calculated via navigation algorithms, with various optimizations, such as the shortest, fastest or cheapest route. In order to implement a navigation algorithm, the navigable space needs to be converted to a graph, compromised with nodes and edge, along with its topology. In a study of shortest path algorithms, Lu & Lai (2006) confirmed the robustness of Dijkstra's shortest path algorithm, even though the shortest routing depends on the structure of the road network. The paths that are possible in a building are not much different. Even though there are no large physical entities hindering, such as rivers or mountains, the buildings structure influences the travel time. A building with two dead ends like structure does not contribute to a short path availability. See also annex B.1. for background on pedestrian movements.

6.4. Navigation infrastructure

For indoor navigation utilizing Wi-Fi fingerprinting as a positioning technique, it is required there are multiple access points, or Wi-Fi routers, are available. In the study area (the OTB building in Delft), no changes will be made to the placements of the routers, as in reality, not always precise coverage occurs, despite it is desirable to do so. Each access point is designated with a Media Access Control (MAC) Address, thus having unique transmitters. Those can be used to determine the position.

6.5. Spatial referencing

A spatial referencing system is needed to properly maintain the locations of the various entities. “Room 329” might be on the third floor in an educational institute, but in a large, single level hotel complex this is not the case. Positioning can be done absolute (positioning of the user is relative to a reference position) or relative (user depends his location on other objects nearby) (Kolodziej & Hjelm, 2006).

In the outdoor navigation, a simple absolute referencing longitude-latitude is sufficient for the user, although the user would prefer a relative referencing system, such as a street with a house number, or a system which supports map backgrounds in which the user can recognize objects. Especially in the indoor environment, the 3D aspects should not be neglected, which is why a position accuracy of 2 metres should is desirable. Multiple objects can be on the same horizontal dimension, yet on a different vertical dimension. Proper geocoding (x,y,z) is thus necessary for each space. With this context, cell ID positioning (absolute) as shown in Figure 9A (page 34), is most likely. A coordination or reference system can be added to each access point, or to each floor, as access points on the same floor usually have the same z-value. The only thing that is required is a switch between the reference systems. However, cell ID positioning is rather inaccurate, since it does not tell where exactly the user within the cell is located. Reference positioning in the form fingerprinting as described in the previous chapters, is a welcome solution instead, since the accuracy is higher. Even though the positioning algorithms are not yet at the desired shelf-like level of accuracy, the deviation of two meters is still considered to be useful, as the user is still able to navigate properly.

Kolodziej & Hjelm (2006), mention that in reality, relative referencing is better understandable. Going right or left is in a humans mind easy to understand, unlike going west or east, as one usually does not have a clue what the orientation is. The same applies to the positioning: *building A, wing B, room 2.10 (second floor)* prevails over position (35.629535, 139.879818; Figure 11) in the human perception. In this context, it is possible to make a set of geocodes from the components of the body, so they can be distinguished from each other, but it is required to recode them again for the end product. This means that spatial referencing will be mostly done relatively.

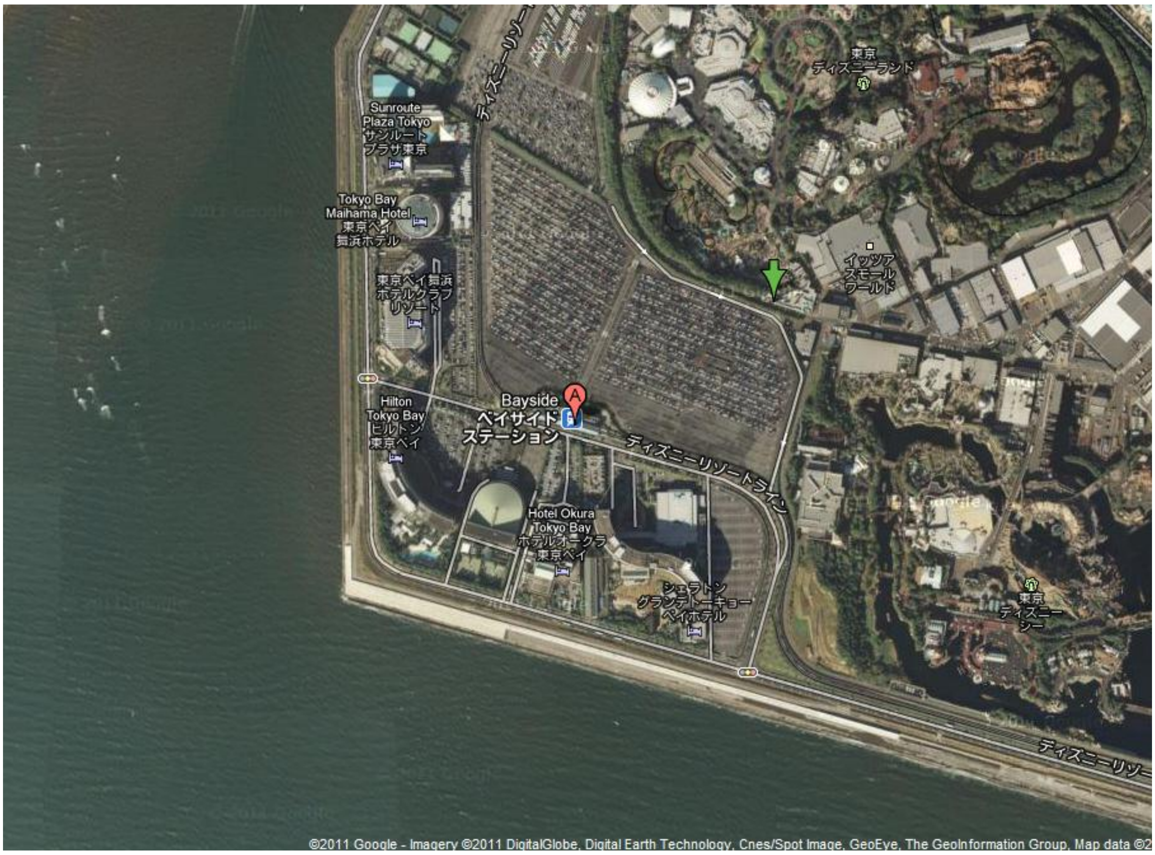


Figure 11. Position (35.629535, 139.879818): Bayside Station, Tokyo, Japan

Source: Google Maps (2011)

The signal strengths of the fingerprinting can be used as a standalone, relative reference system. Each fingerprint has its own location. A position can then be assigned to that location. A reference location (0,0,0) can be assigned to one corner of the building, or perhaps even a little bit outside the building. The exact locations can be recoded to relative positioning parameters, such as <building>.<wing>.<floor>.<room>. Since the goal is to allocate a user using this method, we speak of localization instead of positioning (see Terminology). More about not using an absolute reference system (and geometry) is being discussed in chapter 12.

7. Building an application using standards

An application will be made on a smartphone, which should be able to connect via 802.11/Wi-Fi in the first place. The application shall be installed on a HTC Desire Z device, which uses the Android 2.2.1 Operating System. It is possible to build an application according to existing frameworks. In this thesis research, the aim is to make use of the framework of the Open Location Services (OpenLS) from the Open Geospatial Consortium (OGC).

The OGC has standards for mobile applications which uses a GeoMobility Server, which is a service on a server, where applications can request various services and obtain responses which can be used in applications. The GeoMobility Server is capable to produce maps and routes, as well as accessing other databases via the Internet. GeoMobility makes use of the following five core services (OGC, 2008a):

1. Directory Services (what do you want to find, and where is it located?)
2. Gateway Services (the localization of the user on the service/network)
3. Location Utility/Geocoding Services (the transformation of a location in readable parameters, such as from address to longitude/latitude in decimal degree and vice versa)
4. Presentation Services (the visualization of the results on the device)
5. Route/Navigation Services (the navigation from origin to destination)

The parameters and attributes of these services are described in the OGC document *OpenGIS Location Services (OpenLS): Core Services*. However, most of these parameters require the presence of geometric maps and (by default WGS84) coordinates. As maps are not present in the intended application, only the framework of the GeoMobility Server OpenLS can be useful, in the sense they provide a structure on which services should be present in an application, and how they are interrelated: the same services can be utilized, but with different parameters.

For navigational purposes, the OGC (2010) recommends that at least three factors should be present in the model, and therefore the application:

(1) localization technique and localization infrastructure, which has been discussed in the chapters 5 and 6,
(2) mode of transport (walking; elevators only etc.), by using user profiles (to be discussed in chapter 10) and
(3) navigation constraints, partly resulting from data and modelling issues, as well as spatial referencing, which has been discussed in general in chapter 7, and will be further discussed in chapter 14.

From these three components, the navigation engine can be carried out.

In the next part of this thesis research, the Open Location Services are being described in more detail, along with how it shall be implemented in this project.

Conclusion Part I

How can Wi-Fi be used in mobile indoor navigation?

The IEEE 802.11 technology is synonym to wireless internet, also called WLAN (Wireless Local Area Network) or Wi-Fi (Wireless Fidelity). The technology makes use of the 2.4 GHz Industrial, Scientific and Medical band, and uses spread spectrum modulation. In the transmittance of Wi-Fi, multiple layers are being broadcast. For Wi-Fi fingerprinting, only the header of the power management layer is necessary, which contains information about the header information, and the signal strength. Wi-Fi is already widely available and installed in many public and private buildings.

Fingerprinting is the process of position determination, using the received signal strength at one point. This can be stored for each point, at given heights, although in this research only the height of about 1 metre above the floor is considered. Once collected in a database, the application can check at which position the receiver is, and thus position the user. A major drawback of fingerprinting is that it requires extensive surveying, and that multi-paths works too efficient, since human bodies and closed doors are intervening the path of the signal strength characteristics.

An important aspect of navigation is the confirmation of the user being at the route. If the user gets confirmation, the user can continue to follow his route. If not, an alternative route needs to be calculated. The fingerprints can be considered as check points, in which this thesis will elaborate in the next chapters.

How can an existing infrastructure be used and modified for indoor navigation?

Indoor navigation is different than outdoors, since pedestrians usually do not vary too much in speed, making the shortest path also the fastest path. In addition, absolute positions such as (long, lat) are unsuitable for indoor navigation, as the level of detail is different, although these details still can be used for maps once the indoor locations are being geocoded. New reference systems with a (0,0,0) can be used at the background, while they remain unknown to the user. Relative reference systems are for better use, such as <building>.<wing/section>.<floor>.<room>, in a logical order and sorted. These differences from outdoor navigation are the most important directives for indoor navigation. We then speak of localization, as we use a relative position, instead of an absolute position.

Applications can be built based on the Open Location Standards (OpenLS) set by the Open Geospatial Consortium (OGC). A navigation service can be developed around directory services, gateway services, geocoding services, presentation services and navigation services. In this research, the gateway service is strongly connected with the Wi-Fi technology and fingerprinting positioning. Information about what is available in the indoor environment can be found in the directory services, and geospatial queries can be combined with it, as soon as that information is geocoded according to the geocoding services. The navigation service can thus start if these are combined. The final visualization or presentation is being described in the presentation services, in which another important aspect needs to derive from outdoor navigation: indoor navigation requires much more detail. Hence, a different approach is needed for the final visualization, especially since the navigation also comes on a small screen, where normal cartographic rules differ from large screen maps or paper maps.

PART II - Theory and application

Contents

In this second part, theory and application are made concrete. This part focuses on the application of theory within the to be developed prototype. With the Wi-Fi technology and fingerprinting methods, a navigational application can be created, with the use of the OGC Framework for Location Services.

First, in chapter 9, one must know what the target user is, what one can do and what one can expect. As soon as these user requirements are known, one can further develop the application. In chapter 10, the link between the sensor technology and the application is made. In terms of OGC Location Services, this includes the gateway services. Thus in chapter 11, the directory and the geocoding services are addressed, to make sure fundamental information can be queried and stored. It is important where the points of interests are. The uniqueness of this project is that it does not use building geometry for navigation and the indoor modelling. Chapter 12 emphasizes this aspect. Chapter 13 continues with the presentation and visualization of the before mentioned contents. Chapter 14 deals with the core service of navigation itself, and its extensions. Finally, chapter 15 altogether displays the architecture, which includes the functional and technical design, emphasizing all chapters.

In the end, the research questions *how can the creation of an application be possible, without using geometry?* And *what are the benefits of using the framework of Open Location Service standards in an indoor navigation application?* can be answered.

8. The user involved

Upon building an application or product, it is important to know what your target user is, and what the user would like to do with it. For instance, it does not make sense to provide directions along motor highways if your user is a pedestrian. Creating an application is not any different. This is why there is a variety of location based services available on the market, which utilizes various sensor technologies: from trackers to navigators, from compasses to tagging software. It allows applications to become flexible, as it adapts itself to the user preferences.

8.1. User profiles

It is also possible to install a module in an application that automatically recognizes the user's habits and characteristics each time the application is being used. The aim of such automated modification is to serve the user by supplying information that will most likely pique the interest. Examples are the customized ads in Google's Gmail, where advertisement is closely related to the user's e-mail contents. This is being described as user profiling. Wu et al. (2009) describes this as "the characteristics and the relationships of the different users" being used, although the European Telecommunication Standards Institute (ETSI, 2010) better describes user profiling as a module "which formally captures the user requirements (preferences) […] and deploys those profiles […]", covering explicitly users desires. They are going even further, by defining user profiles as a "total set of user-related information, preferences, rules and settings, which affects the way in which a user experiences terminals, devices and services". User profiles are different from user requirements, as user requirements involves the desire of user in technical fashion in the development of the product, rather than the adaptability of user profiles in the actual usage of the product.

8.2. User needs in this project

In this project in particular, the goal is to guide a person in a public building with its mobile device from the entrance to a given destination.

The aim in this project regarding user profiles, is to develop such module within the application at a simple level, *id est* storing the information that is being inserted, so that in future events the application is able to predict what settings the user most likely wants. From the final definition the ETSI gave, (1) personal information, (2) human centred preferences, (3) service related information and preferences (specific information-feed for the service) and (4) device related information and preferences, all belong to the user profile. Ultimately, it is desirable to create an application that is able to read device information, making the application suitable for a multitude of platforms, but in this project this will be limited to the capabilities of the Android 2.2.3. HTC Desire Z device, since it is the test bed. This results in the focus of user profiling in the user personal information, preferences and service preferences.

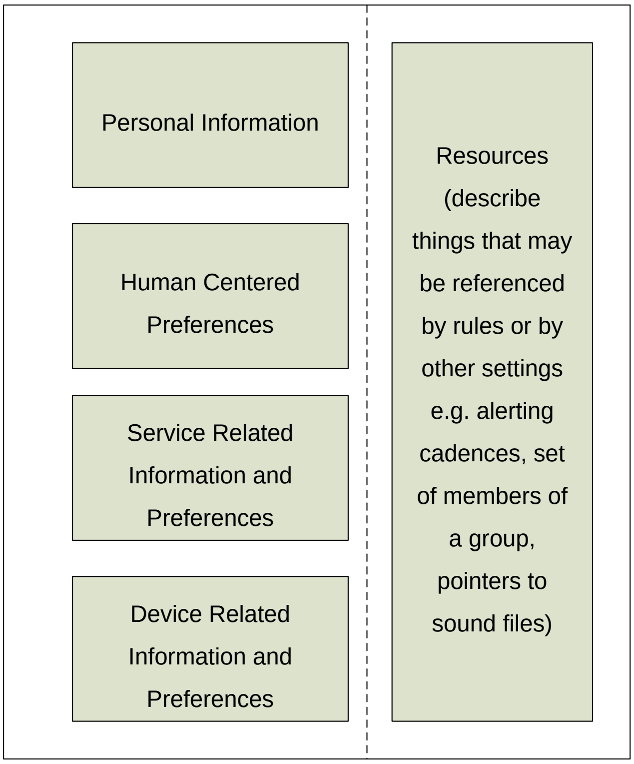


Figure 12. User profiles defined by ETSI (2010)

8.2.1. Personal information

In this project, the person is modelled with the following properties (see also Usher & Strawderman, 2010, for extensive research on characteristics on pedestrians):

- The user has limited information about the indoor environment.

This means that the output of the application should compromise clear descriptions and overviews about what to do.

8.2.2. Human centred preferences (requirements)

The following characteristics will co-determine the application regarding the preferences of the user

- The user is provided with a simple overview of how the application should work.
- The user wishes to get directions to walk towards the destination.
- The user wishes to get confirmation whether the user is on route or not.

8.2.3. Service related information and preferences

Combined with the human centred preferences, the service related information and preferences aims at the possibility to adapt the application to the user. This means that:

- The application allows a customizable feature that influences the whole concept.
    - Or: that the user can change settings of the application and adapt it to its situation.

8.2.4. Device related information and preferences (requirements)

Finally, the device related information and preferences are coping with hardware constraints. The application needs to fit accordingly:

- The user uses a HTC Desire Z smartphone model, with Android, API level 8 (2.2 or higher).
- The application should run on a platform with fast processing (short processing time).
- The application allows a touch screen interface.
- The application utilizes Wi-Fi technology and must be enabled.

An example of using user profiles is illustrated by Foerster (2010) about map usage: a map of a specific region is the same, but the contents being displayed for a geologist is different than for an urban planner. The map is adapted according to the user. Some background information about maps being adapted on mobile devices has been provided in Annex B.2.

9. User allocation: gateway services

Differences between indoor and outdoor navigation includes the sensor technique (GPS versus non-GPS) and the scalability: the destination in the outdoor world are usually points of interests, while indoor navigation could be seen as a subset or complement of the outdoor world, meaning that a single point of interest (building) possibly contains other points of interests as well. These points of interests need to be carefully maintained, since a large amount of attributes can be coupled to these points of interests, which can help the user to navigate inside buildings.

The localization of the user in the indoor environment can be further used and integrated in location based services, which is being described in this chapter. Further details about the building itself needs to be mapped, and the rest depends on what the service provider would like to offer (Jan *et al.*, 2010; see also the remaining chapters of this part II).

9.1. Components of the gateway

The positioning and point placement of the user to the location service (the gateway service) is perhaps the most challenging part of the project, as different algorithms applies to Wi-Fi positioning. Kolodziej & Hjelm (2006), propose the following steps need to be done, in order to secure a decent gateway service:

- Site survey performance (signal strength model / fingerprinting database)
- Creating the positioning model
- Calibration
- Access point placement and configuration
- Tracking
- Maintenance
- Rights (due to changes over time)

Since it is the aim of the thesis to make a working prototype with existing access points, only the site surveying is being discussed. It is not the intention to fine tune the access point placement, since the main research question is whether it is possible to develop a Wi-Fi fingerprinting application without using geometric properties.

9.2. Site surveying and positioning

Site surveying is the preparation phase of fingerprinting. A database will be constructed with the specific fingerprints. It is possible to conduct this phase in the three dimensional space, by measuring in vertical positions as well. However, since most people hold their phone around the 1 metre height, only measurements at this corresponding height take place, at each floor.

The temporal aspect has to be acknowledged, since the fingerprint can change over time, because a receiver can obtain signal strengths directly via a line-of-sight, but also indirectly, via reflectance (the multipath). The process of measuring has thus to be repeated at various time slots.

The specific fingerprints will be assigned to a “label”, which actually includes other information about what is present at that place, such as recognizable objects (plants, pillars, inboxes, and signs). This process is similar as illustrated in Figure 10 (page 35). In this thesis, position is considered synonym for a fingerprint, since the fingerprint can contain already information about its environment. Real map matching and geometry are not required as such. It is therefore not possible to create a positioning model. In chapter 11.2, more background is given on the non-usage of geometry.

10. Directory and geocoding services

The reason why these two services are in one chapter is that they are closely related to each other, in the sense they are both concern information about data: the directory services provides information about a place or object, whereas geocoding services provides information about what the location is.

10.1. Directory services

The OGC (2008a) describes a directory service as [a] *service [providing] subscribers with access to an online directory to find the nearest or a specific place, product or service.* This implies that a user can use a search engine to find something with or without explicit location specifications. Users might find a place, product or service by providing specific or explicit properties, which is called pinpointing. Alternative, they can use spatial queries to find the nearest place, product or service, called proximity. Both pinpointing as proximity require the following minimal information:

- A property of the point of interest, such as a unique id, name, description, absolute location
- The relative location information to the point of interest, such as *nearest to me*, *within 500 metres.*

The OGC (2008a) provides specific directory request parameters. The only mandatory parameters include (although this depends on the nature of the query, in which other values can be stated as mandatory):

- POIproperties, in which a listing is provided, that can be used to do the search (keywords)
- Name, in which characteristic values are stored, such as the reference object, the name and the type.
- Value, in which the information can be reflected to the user.

An example is illustrated in Box 1, in which the question is asked where the nearest copy machine is being searched, while making use of a relative location statement.

Where is the nearest copy machine of where I am (Room 1.002)?

```
<DirectoryRequest>
        <POILocation>
                <Nearest>
                        <POI ID="1">
                                <POIAttributeList>
                                        <POIInfolist>
                                                <POIinfo name="POI Name" value="Room 1.002">
                                        </POIInfolist>
                                </POIAttributeList>
                        </POI>
                </Nearest>
        </POILocation>
        <POIProperties directoryType="Campus">
                <POIProperty name="type" value="Machine">
                <POIProperty name="subType" value="Copy">
        </POIProperties>
</DirectoryRequest>
```

Box 1. A XML Example for OGC Directory Services

The provided output is list of points of interests with the distances.

This assumes that upon developing the application, at least a database with retrievable information for the interest of the user should be populated.

10.2. Geocoding services

In order to make locational queries possible, it is necessary the various points of interests needs to have information on their position, also called geocoding. The other way around is called reverse geocoding. The geocoding is usually carried out on the background, since the services need to know exactly where the point of interest is located, and the user usually does not know on which absolute position he is. As soon as the point of interest has been traced and located in the absolute reference system, it will be translated back to the user in named locations, such as *Aisle 1*.

For geocoding services, the following parameters are mandatory:

- Address, which can be inserted →the returning value would be a position (X,Y,Z)

For reversed geocoding services, the following parameters are mandatory:

- Position (X,Y,Z), which is the starting position of the user which can be inserted, in terms of absolute positions. → the returning value would be an address (street, number)

In our iindoor environment, it is unlikely one can insert position information on its own, since a fingerprint is being used. Unless the user knows exactly what fingerprint he or she is going to insert, it is not possible.

The returnable responses for geocoding services that are possible include the following:

- numberOfAddresses, so that the user might choose from matching addresses, if there are multiple matches (e.g. if one inserts “Commission Room”, it is possible that there are 3 “Commission Rooms”)
- Address, which matched the inserted address.

11. Modelling and attributes

11.1. Modelling the indoor environment

The indoor environment can be modelled in many ways, dependent on what the service is aimed at. The importance is that one must know the level of granularity should be. As it is the goal of this project to let users navigate in buildings to a room, the accuracy level should be *a room* and not *a part of a room* (shelf-like accuracy), unless the target is a very large space. A node and edge structure like has been proposed by Kolodziej and Hjelm (2006) and the OGC (Open Geospatial Consortium; 2010) to model the environment for indoor navigation. Especially the OGC highlights the importance of the relation between the nodes and edges, especially in the larger rooms and corridors. It is therefore recommended to break larger rooms in smaller segments; the OGC calls this partitioning. The vertical dimension can also be part of the model, as shown in Figure 13. In fact, this way of modelling resembles of remodelling the topological dual space as the node-relation-structure (OGC, 2010), which can be used for the navigation service. The dual space can be described as a model of the space in terms of linkages, which resembles a graph. The topological dual space explicitly preserves the topology of the graph.

It is hard to tell whether rooms are neighbouring, since there is little information about the size and distance of the environment, unless extra information can be added to the graph. Therefore, attributes can be assigned to both the node as the edge. Those attributes can be used for locational queries, in which the user can start its navigation in the first place. This also includes non-navigable spaces for specific users, such as bumps for wheelchair users, which in turn can be used to determine alternative routes.

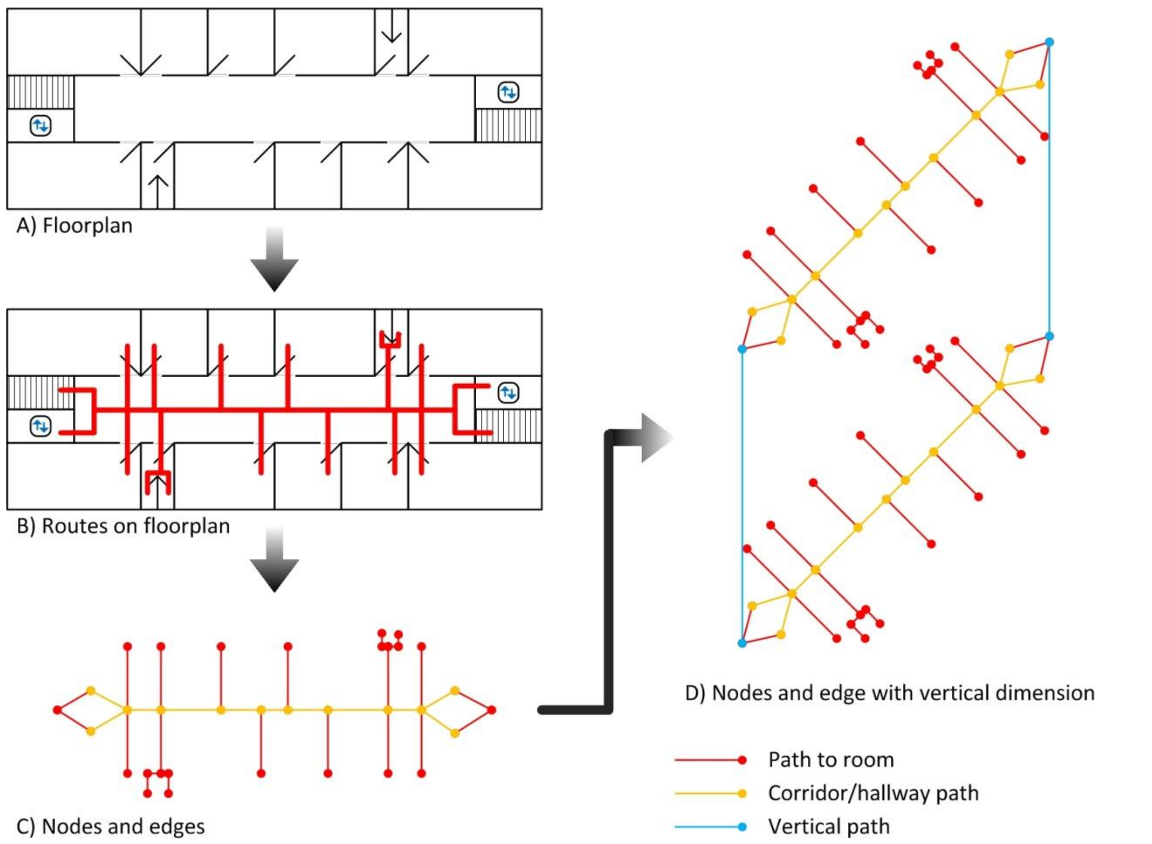


Figure 13. Nodes and edges representation (topological dual space)

11.2. Non-usage of geometry

Unlike the outdoor navigation, data on indoor locations is not widely available. There are no vendors for networks such as FalkPlan and TeleAtlas, and not all companies are willing to provide floor plans for public use, or even if there is a provision, it might be possible that not everything is covered to keep it secret. This might cause less accuracy, or missing data, although this might change in the future.

In fact, the geometry does not necessarily have to be used in order to deploy the navigation service. Indoor, a user only needs to know whether he is still on route, going from A to B. Going on route, there are significant recognizable objects present, such as plants, pillars, postal boxes or paintings, which provides help to confirm the user is in the right direction. For this, only fingerprints are an absolute necessity. The requirement is that the fingerprints should be distinguishable from each other and attached to POIs. Those fingerprints can be stored in a database and can be coupled with attributes (recognizable objects) about what is present at that place and organized in the graph. These attributes can be displayed on the screen in text or image. If the user can see those objects, as the screen tells them, the user can be certain he is at the right place. If the user reaches a node where a vertical movement can be made, information about possible stair climbing can be returned as well. This whole process is illustrated in Figure 14.

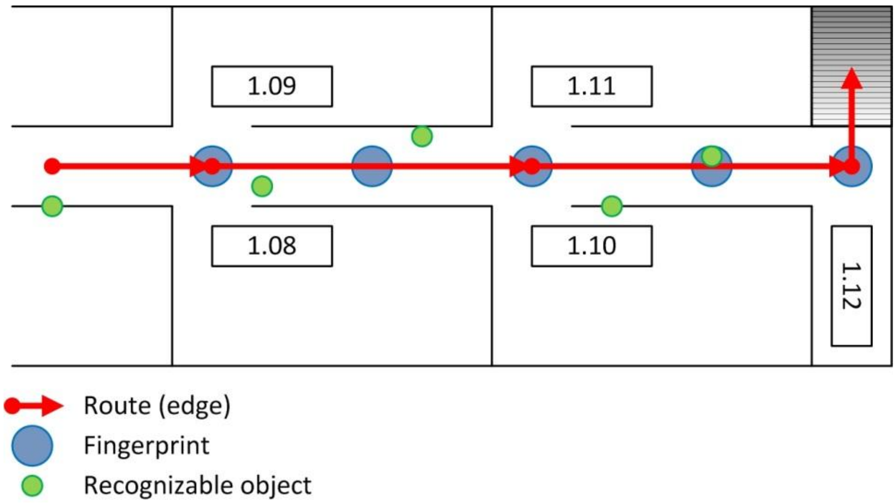


Figure 14. Fingerprinting objects and navigation

The above implies that there is no geometry necessary, but this means that topology is indispensable. As indoor pedestrian navigation (see chapter 7) has the opportunity to recognize the space in multiple ways through present objects, it is not necessary to plot the entire floor plan: a fingerprint is sufficient. The only information the pedestrian can have, is how to go from one end to another. However, future extensions include the overview of the graph, so that the user can see at least the route on screen. Another extension is the estimated time of arrival, which. However, the travel time variables significantly, as this depends on the user profile. Some users need more effort to open a door, while others need more effort upon climbing a staircase. This suggests that only information about the edges is necessary, and not about the geometry surrounds it, as long as the topology remains. The result is to be found in Figure 22, and is discussed in chapter 16.

12. Visualization: presentation services

In this chapter, the visualization and representation aspects are discussed. As standard cartography deviates on mobile platforms, first generalities on small screen cartography are discussed in 12.1. In 12.2 some technical information on visualization is related to the OGC standards on presentation services. Finally, in 12.3, some comments are made about 3D visualization and explained why this is not part of the thesis.

12.1. Users and general constraints and limitations

A decent visualization is necessary to communicate the proper information to the user. The user is most likely not interested in the underlying principles, but in the end product. Small mobile screen require a different visualization than paper maps or large computer screens. Nagi (2004) distinguishes constraints on both the device-end as the network ends, as displayed in Table 4.

Category	Infrastructural constraint	Constraint	Comment
Display	*Device*	*Screen size*	*Usually very small*
Display	*Device*	*Resolution*	*Usually very limited*
Display	*Device*	*Colour range*	*Low range, although depends on device*
Interaction	Device	Input opportunities	Limitations of input fields, although depends on screen & resolution
Interaction	Device	Output capabilities	Usually restricted due to limitations in processing
Performance	Network	Connection	Slow in contrast to regular (wireless) internet, unless device has a Wi-Fi data connection
Performance	Network	Latency	
Performance	*Device & Network*	*Limited storage*	*Newer generation mobile phones allows larger capacity with the expansion of (micro-)SD-cards*
Performance	*Device & Network*	*Processing power*	*Restricted in the sense it does not equal normal computer-like processing speed*

Table 4. Mobile phone constraints

Table is based on Nagi (2004).

Although the information in Table 4 is dated from 2004, with performance and display related constraints being partly solved or relieved. Still, the display and interaction categories remain important limitations on the end visualization. The cartography largely depends on the displaying capacities of the device: the larger the scale of a map, the more detail is needed. On a small screen, this is considered challenging to overcome this problem. Yet, the better the resolution, the better the application can overcome this challenge. A low resolution smartphone might display text labels or symbol too large, for instance (Nagi, 2004). Newer generation smartphones are able to scale the display; in that sense this barrier has overcome.

With the use of zooming options, one can toggle through different levels of details; however this usually takes a lot of processing, when dynamic zooming is being used. This means that for each scale a different data set and data representation is being visualized, which is a smooth zoom in fact. This can also be done in steps (static stepped zooming) or static linear (no new data sets being visualized at all). Either way, this demands a lot of processing time for both the device as the network (Nagi, 2004).

12.2. Presentation services by the OGC

The presentation service of the OGC (2008a) makes use of Web Mapping Services (WMS) and Web Feature Services (WFS) to display on demand maps. However, since there is no geometry, only static figures (images, topological network graph without coordinates, maps) can be visualized instead, as there is no active and adaptive map viewing system. The parameters which belong to the presentation services are all not applicable for the purpose of this thesis research.

12.3. Acknowledgement of 3D modelling and 3D visualization

Although developments of 3D modelling and 3D visualization are still ongoing, it is decided in this research the visualization remains on the 2D or the 2.5D plane, as the focus is on the confirmation of positions, and not the visual representation. However, the acknowledgement of 3D visualization is important in future enhancements on the screen. Even though the presence of flyovers or overpasses in the outdoor world is missing indoor, pedestrians still needs to deal with vertical movements along staircases and elevators inside buildings. Döllner et al. (2007), suggests that 3D visualization of objects or buildings certainly improve recognition patterns of users, which will further improve the navigation process. It is not necessarily to make photo realistic representations, but 3D sketch-based commands are already sufficient. In addition, utilizing the camera function of smartphones, augmented reality (placing layers of information on top of the visible environment) belongs to another possibility which can be incorporated in the application.

12.4. Cartography of text

The well-known outdoor large points of interests in the form of buildings do not apply to the indoor context, however. Instead, smaller points of interests are present. Simple room destinations can be translated to concrete goals and information. For example, room 1.001 can be an information point for administration, and room 1.002 contains a copy machine. The visualization for the indoor navigation follows thus the same rules and relations as described in the previous two sub sections.

However, as a result of not using the geometry for the modelling (chapter 11), no maps can be displayed. It is possible to provide the user with a graph or with simple listed text to direct the user, instead. The necessity of a map thus disappears, although this assumes there is a significant amount of recognizable objects present. The presentation of the service could be as follows (Box 2):

```
1. Keep straight as you pass a copy machine
2. Keep straight as you pass inboxes
3. Keep straight as you pass room 1.011 to your left.
4. Move up one floor.
```

Box 2. Example of utilizing text for recognizable objects

The advantage of no cartography (maps) means that time and effort can be saved for deploying the application on a larger scale, as one only needs to determine the fingerprints. As Hsu (2001) points out, text usually takes longer to process for the user, which can be considered as a disadvantage. Ideally, the text should be as short as possible, and supported by icons. The sub sections below will discuss the underlying principles of text usage and text visualization as a whole.

Text visualization

From the general cartographic rules, special attention is provided to the text visualization in this paper first. Although Kraak & Ormeling (2003) only describe the rules for text usage on maps, the same rules can be used for general text emphasis without maps. Text is able to show levels of importance between words/names, by creating variation in:

- Highlights, using boldness or italics;
- Font face/style, using different fonts to distinguish texts from each other;
- Font size, to emphasize the (non)-importance of words;
- Font spacing, to emphasize the (non)-importance of words;
- Font colour and/or font colour background;
- Upper and lower case, by creating a contrast on what should be read.

Again, colour can be used to differentiate text on maps, but also on non-maps. The same limits on colour perception apply: a maximum of approximately 7-8 colours can be used to ensure the reader is able to select the most important information from a line of text.

Text attractiveness and uniform design

The screens on mobile phones thus has limited space, and although the scaling on screens have been improved over the last years for smartphones, the emphasis on information visualization remains a significant contribution to an application, aside from what the application does. Roto & Kaikkonen (2003) found out that the topmost information on a page or interface got the highest priority, since it is first item perceived. In a study on the user perception of web pages' interface, Hsu (2011) researched the attraction of elements on a webpage and listed them from best attractive element to least attractive:

1. Title/Logo
2. Image
3. Information display/presentation
4. Willingness to read
5. Colours
6. Structure
7. Attraction
8. Layout
9. Usability
10. Hyperlink
11. Readability of texts

As can be seen, the importance of text is rated low. Suggestions on images, colours and layout are better rated in contrast to texts (rank 10 and 11, aside from the aspect *willingness to read*). However, when it comes to the functionality, texts were provided more important; better than the title or logo. Text has to be readable to obtain knowledge as such, and Hsu (2011) found evidence that the texts are in general better perceived when the layout of the text follows the webpage or application standard design, which has to be uniform. This does not differ from location based applications.

Variation in text elements in their context

Michailidou et al. (2008) also underlined the aesthetics of text visualization in webpages and applications. They acknowledge the importance of using clean, clear, and symmetrical design, in which bullets and buttons can further enhance the readability of text. As with the general cartographic rules, further refinements can be shown, using text variation in colour, size and style. Wang & Chen (2003) provided evidence that important text can be distinguished within the main text, by placing extra space on both sides, in which they call this jump length. A minimum distance of 0.35 cm to 0.70 cm has been considered effective – if the distance is too long, the text would appear to be discontinued, in which users perceive as disturbing (too large holes in the text).

Text colour usage

Wang & Chen (2003) researched in particular the combination of colours on screens, for both texts as backgrounds. They stated that colours have to be carefully chosen when they need to be combined. Two extremes of the colour spectrum have to be avoided, such as red and blue, as they provide the user from a “visual discomfort”. More important is the luminance contrast between background and text colours, as they are perceived much better. For example, the combinations of black-on-white, white-on-black and blue-on-yellow were better read than red-on-white, blue-on-white and green-on-white, although the combinations are dependent on the condition of the screen, as well as the suns illumination outdoors.

Moving text

Wang & Chen (2003) also investigated the effect of using motion and speed of text, and they found that moving text in application has little impression on the user. In fact, users were tempted to ignore the moving text, which has also been acknowledged in a study by Albrecht-Buehler et al. (2005). Moving text in applications should not be too slow, but not too fast either, in order to maintain the attention of the user. In combination with background and colours, one has to ensure the background should not be disturbing to read the texts themselves. If the motion is too slow or too fast, or if the background and colours makes it hard to read, change blindness (ignorance of texts, thus ignorance of information and ignorance of the application) is being the case, which has to be avoided.

12.5. Initiatives to a text based indoor navigation application

Based on the above sections, an indoor navigation application, which is based on a small screen, and does not utilize geometric maps, but solely text display for directions, a few recommendations can be made regarding the output:

- The interface must be adaptive to the user.
- The interface itself has to be uniform in layout.
- A contrast in colours (luminance) has to be guaranteed, in order to let the user perceive the information correctly. Blue-on-yellow has been proposed.
- To the top of the screen, the most important information has to be displayed.
    - This coincides with the current location and the target destination.
- When it comes to the output of text directions, orientation aspects have to be distinguished from the body.
- There should be a limit to the motion of text – text scrolling should be made possible otherwise.
- Text size can be considered as well.

The proposed indoor navigation application is as follows:

1. The device automatically detects where the position is.
2. This position is made visible on the top of the screen.
3. The user has to insert a specific start or current location and destination.
4. The user has to navigate from one end to another.
5. The device calculates a route.
6. The user sees his current position on the top of the screen.
7. The user sees if he is on route/off route directly below that position information line.
8. The user sees his destination and the estimated time of arrival.
9. The user sees a list of directives below that on/off route information.
10. The first line of the directives is placed in a bigger text size.
11. The subsequent lines are in a different colour and size, to emphasize the importance of the first direction output.
12. In any of the lines, information about directions can be made in a separate colour or style.
13. In any of the lines, information about orientation objects can be made in a separate colour or style.
14. Bullets or numbers have to be used to indicate steps.

For example:

Current location: near room 5.99
ON ROUTE | Heading to: Exit
- Coming from the stairs, walk **straight** towards Room 5.97 (to the left you see glass works, to the right you see a coffee machine)
- **Turn right**, you see an elevator
- **Take the elevator** to **ground floor**
- From the elevator **turn left**
- **Exit** the building

Figure 15. Example text output using text visualization and variances

13. Navigation services

The navigation engine is another vital part of navigation as a whole. The OGC (2008b) enhanced their route services with the navigation services, with the assumption that the user is looking for more than solely a route, such as the reconfirmation of their progress, and editable features.

A brief description of the Navigation Services from the OGC is in section 13.1. Section 13.2 deals with the mandatory parameters to be used in a possible Navigation Service. The full overview can be found in OGC (2008b). Section 13.3 concerns with the generation of route descriptions.

13.1. Navigation elements of the OGC

The OGC (2008b) listed 30 possibilities determining or influencing the course of the user, in which the following are considered to be the most important. The full list can been found in Table AP. 2:

- Core navigation:
  - Specification of route preferences
  - Specification of origin and destination point
  - Determination of position
  - Route calculation using user profile/preferences
  - Presentation of route
  - Route guidance
- Extensions of rerouting:
  - By detour, with traffic information
  - By waypoints, as defined by the user
  - By getting off route
  - By change of route settings and preferences

In this thesis, getting off route piques the most interest. This process assumes that the user have to be tracked, to determine whether the user is on or off route. Thus, the confirmation of the user's position plays a fundamental role in this aspect. In the case of fingerprinting as a positioning technique: routing is made possible by linking the fingerprints with each other and the user has to pass each point (fingerprint) on the route to stay on route. The navigation client should tell whether the user is on or off route, by comparing those fingerprints with each other.

It is not the aim of this thesis to extensively describe all parameters and specifications of the navigation application itself. The OGC (2008b) provided all exact inclusions in the document of Navigation Services already. In the next section, only the mandatory parameters are briefly discussed.

13.2. Mandatory parameters

The OGC (2008b) states that *"map matching is the method of determining where the "mobile device" has moved in the navigable network based on the device's previous location(s) and data about the device's motion from external input (such as, but not limited to GPS)"*. Map matching is strongly related positioning, since a position is needed in order to determine where the user is. The OGC (2008b) describes recommends hard lat-long coordinates in the WGS84 reference system, but this cannot be implemented for the interior of a building. Instead, fingerprints in the context of a topological graph are a possible solution.

At specific points on the navigable network, attributes, conditions and relationships (i.e. prohibited manoeuvres and directions) needs to be assigned (OGC 2008b), in order to provide the user directions. This means that entries and exits are also part of this.

In Table 5 below, the most important parameters are displayed. A full overview can be found in the document Navigation Services of the OGC (2008b).

	Parameter	Mandatory	Short description
Route requests	RouteHandle	Yes	Request information about the route
	ExtendedRoutePlan	Yes	Specifies criteria route determination
	BoundingBox	No	Rectangular area of route
	RouteGuidanceRequest	No	Return of turn-by-turn route instructions (text)
	FirstBucketSize	No	Stores directions in a list (“bucket”)
	Priority	Yes	Priority of requests
	provideRouteHandle	No	Return of a route handle
	DistanceUnit	No	Unit for measuring distance
RoutePlanType	WayPointList	Yes	List of waypoints along the route
	AvoidList	No	List of areas, locations and features in which the route should avoid passing through
	ExtendedRouteControl	Yes	The criteria upon which a route is determined
RouteControlType*	CostCriteria	Yes	Cost Criteria
Route Response*	RouteLinkAndCost	Yes	Ordered list of links and node travel costs for the path
	FirstBucket	No	Display of where the route navigation is now
	RouteHandle	No	Reference to the route stored at the Navigation Server
	RouteSummary	No	Describes overall characteristics of the route

Table 5. Parameters Navigation Services

*Source: OGC (2008b). * More parameters are included, but these include vehicle information and actual traffic information, which does not apply to indoor navigation in general.*

As derived from Table 5, the control of the route, including the waypoints is fundamental for the navigation. In order to navigate, one must include navigable points along the way, and assign costs to it. These costs can be used to inform the user about time and distance estimations.

13.3. Generating navigation guidance

In this thesis research, a simple navigation algorithm, using Dijkstra’s shortest path route, can be used for navigation instead. As the intention of the thesis is to use solely the framework, the parameters in table 8 have to be transformed in Java (see chapter 16). However, before using the parameters, thoughts on how the route descriptions (the guidance) have to be explained first.

When no maps are preserved, it is challenging to determine orientation aspects. Upon travelling from node A to C via B, it is hard to tell from solely a graph whether one has to turn left, right or keep straight, as one can flip the graph, for instance. It is then important to assign orientation aspects to a connectivity database, when nodes (the fingerprints) and edges (connectivity information between fingerprints) can be stored. It is then important that the application should take the current and predicted positions into account.

When the current and predicted (or the previous and current) position is known, the application can then predict on how the orientation can be helpful. In Figure 16, a route has been calculated from node 2004 to node 2012. Upon travelling from the first nodes, from node 2004 to node 2002 is basically straightforward: it implies that going from node 2004 to node 2003 is straight on, and the user has then turn left. However, there is no explicit implication that this graph is mirrored: perhaps going from node 2003 to node 2002 is a turn to the right. When storing orientation information between two links, one cannot add left or right, since it depends on where the user is coming from. In the inverse route, the turn 2003-2004 is 'right', instead of 'straight on'. It is useful to store information of the current or previous location, the predicted location, or both. In a table, one could store the route 2004-2003-2002, with 2004-2003 straight (previous location was 2005) and 2003-2002 as a turn to the left, while 2002-2003-2004 deals with a 2002-2003 straight (previous location was 2007) and 2003-2004 as a turn to the right.

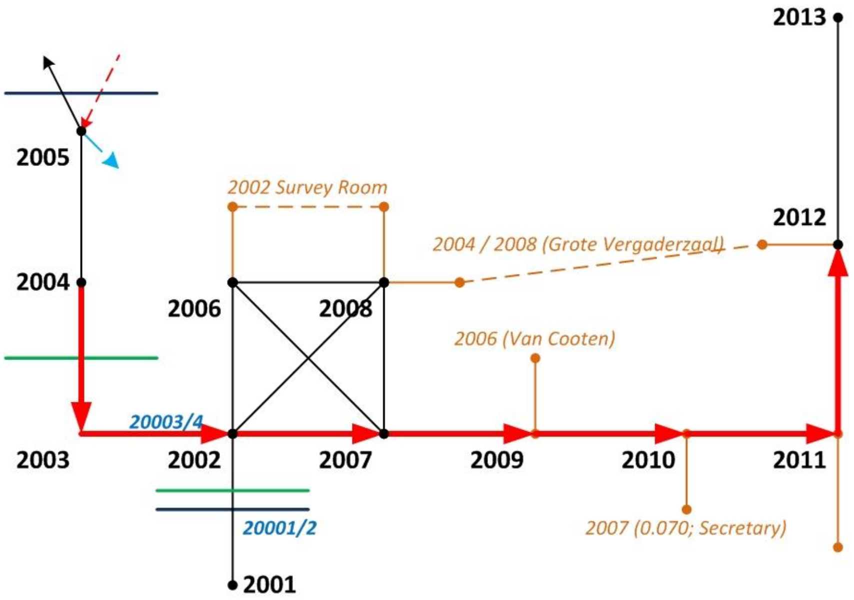


Figure 16. Routing, guidance, orientation

However, there are many routes possible, and the question is how to program the orientation aspect. One possible solution, is by pre-programming every single possible list of route descriptions, as described above. Another possibility is to store unique location, object information, without using orientation terms, such as 'turn towards the floor sign', although it is difficult to state, when there is no geometry involved. It is then important that the object information has to be stored along with the fingerprint or with the connectivity information (or both). In the case from travelling from node 2004 to node 2012, the following can be constructed:

- *(2004) Pass through the doors ,*
- *(2003) turn towards plant ,*
- *(2002) walk towards sofa's ,*
- *(2007) pass the reception ,*
- *(2009) pass Anneke van Cootenzaal ,*
- *(2011) turn towards statue ,*
- *(2012) destination reached.*

In this sense, no orientation (as in: head north/south, turn left/right), or rather, neutral directions are being used. Only in the vertical direction, explicit orientation directions can be stated (go up/go down).

Next, the application has to check the current position of the user constantly. If the user finds itself on or near the provided nodes, then this must be confirmed. If not, re-routing has to be done, in which the user should be warned. At the same time, the application can already predetermine which nodes can be detected. It is therefore useful that the application should only get nearby nodes (fingerprints). For instance, the application should at least eliminate outlying nodes: if one traverses in the route from node 2004 to node 2012, then the application should not detect the user at node 2005, node 3005 or node 3205, after passing node 2009, for example.

In the next chapter, an architectural overview is being presented.

14. Architecture: functional and technical design

With the use of the services and user information, the final architecture can be determined, which is displayed in Figure 17 below. The described core services in the chapters 9, 10, 12 and 13 are embedded in the centre of the architecture, based on the principles mentioned in chapter 11, as they deal with the processing of the application. The application itself is constructed in Java with Android, and is located at the heart of the architecture. The positioning feed is derived from the position determination equipment; in this case, this is the Wi-Fi receiver of the Smartphone.

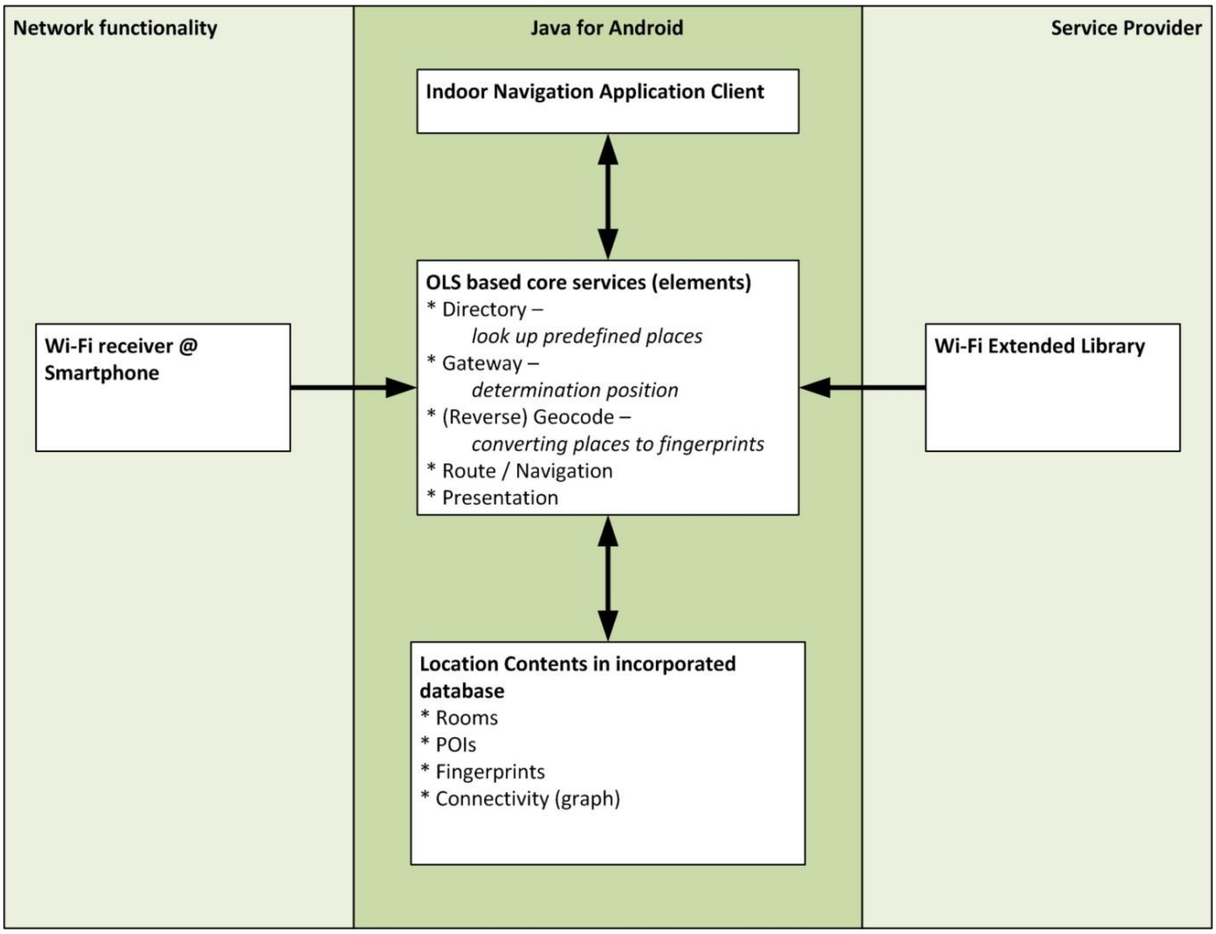


Figure 17. Architecture of this indoor navigation application

Derived and edited from OGC (2008b)

The location contents form the database of the application. This also includes the fingerprints. As the application itself is standalone, there is no server needed. The application constantly links to the inherent databases, which contains information on rooms, POIs, which are combined with fingerprints and their connectivity. The application itself makes use of the Wi-Fi receiver of the smartphone, which is processed by a Wi-Fi Extended library, also in the Java environment. The client (the Graphical User Interface, the GUI) finds itself at the top of the architecture, as it can request the information from the application.

The functional design allows seeing what the user can do with the application. Based on the documents of the OGC (2008a and 2008b) and the visualization described in the chapters 9-14, both the functional as the technical design is displayed in Figure 18. Ideally, only a database with directions and inherent information about the fingerprints themselves are necessary. The fingerprints directly link to the locational information, and the route information.

The functional design is explained as follows (in the figure, in red denotes actions to be taken for the technical properties of the prototype):

1. The user has to make sure Wi-Fi is enabled. In the case the inherited databases are placed on a server, the user has to make sure a 3G OR Wi-Fi connection is enabled, so that the databases can be downloaded. 3G is necessary if the user cannot establish a Wi-Fi data connection.
2. The route preferences have to be stated via the user profile. By default, this has been set to a normal pedestrian, with a guest status. This implies that certain areas are restricted, unless stated otherwise. The user can change this by clicking the Settings button.
3. The user can then define the departure. The current location is being displayed, which helps the user orientate. The user have to define the departure location by typing (selecting) a predefined location.
4. The user has then to define the destination. The same categories are possible as for the origin, as assumed before, the user does not have knowledge about the indoor positioning, nor the addresses (other than room numbers), which makes it impossible to pinpoint an exact location. As the application also does not use maps, it is not possible to pinpoint it on a map.
5. Optionally, a via-point can be inserted. The same categories as the destination can be chosen, with the addition that it can be set as an inclusion, or exclusion. In the case of an inclusion, the route given will prompt the user to navigate to this first set point. In the case of exclusion, the route will take into account it is not desirable to pass this point at all. An example is the exclusion of a certain segment, which is now restricted due to refurbishments.
6. Now, the user can plan the route, by touching the plan button.
7. A list of directives will be provided. Each direction has information about what is visible at that point, so the user can see himself whether he is at the right position or not. In addition, it will be visible if the user's location matches the predicted fingerprint.
8. In a possible extended application, the user is capable to modify the presentation, which contains the text display, speech engine and most importantly, a confirmation setting. This can be done by reserving a section on the screen which shows a green approval or red disapproval sign or something similar. Alternatively, this can be spoken as well. In addition, a graph can be displayed, but this depends on whether a graph will be included of the building. For now, the focus is on text directives. The map can be used, for instance, to display the entire route.

At this stage, it will be possible to still adjust the route preferences, as well as the inclusion and/or exclusion of certain areas.

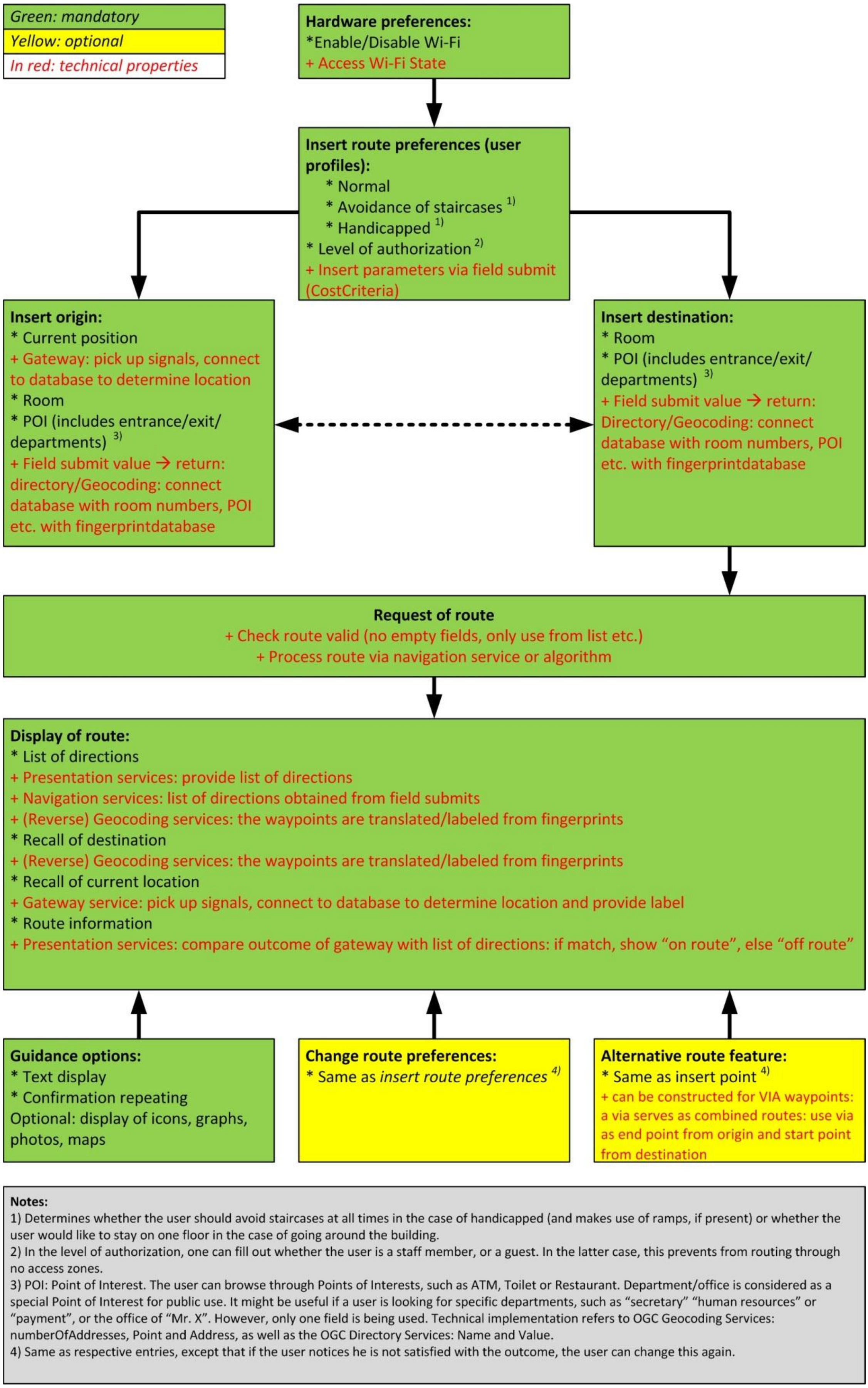


Figure 18. Functional and technical design of the application

Conclusion Part II

How can the creation of an application be possible, without using geometry?

The user would like to navigate from one source to destination. Indoor, the user has points of recognition available, as the level of detail is higher than outdoor. The user can thus check whether a navigation application provides correct information, if the application makes use of the surroundings. As the focus is on utilizing fingerprints only, instead of using geometry, positioning and representation are provided a different context. Fingerprints can be coupled to recognizable objects at that place. A real (x,y,z) positioning (as stated in the terminology) is then not necessary, as user is only interested if he is at the right place at that time, since there are no obvious addresses in the vicinity, with exception of numbered rooms. In addition, pedestrians have specific characteristics, and thus minute estimations are hard to tell, since a user might need more time to climb the stairs or to open a door. This implies that geometry is not required, as long as information about connecting fingerprints is preserved.

As with the level of detail surrounding a user, a simple text list is considered sufficient, as long as it includes information about recognizable objects. However, a weakness is that there is no clear direction or orientation (left/right) possible, since there is no geometry. Instead, neutral descriptions in combination with recognizable objects have to be provided. This assumes that the user constantly has to carefully watch the real environment because of the high detail. In the case graphs or floor plans are provided, one must consider the constraints and limitations of the small screen and processing speed.

How can the use of the framework of Open Location Service standards be implemented in the indoor navigation application?

Each component of the Location Services can be used in the Technical design. The most important feature is the gateway service, where the users location as picked up by the Wi-Fi receiver, has to be compared to the fingerprints database. The user itself can define the constraints and preferences, which determines the route handling. Upon entering information on origin or current location, destination and the inclusion or exclusion of via points, information can be transformed via the directory and geocoding services. This information can be further processed, with the navigation service. However, since the aim of this thesis is not to use geometry at all, large parts of the parameters of the OpenLS standards cannot be implemented. Therefore, only the framework will be used instead of the parameters.

PART III - Implementation results

Contents

In this third part, the final output is being discussed. First, the methodology of surveying as well as the creation of the database is discussed in chapter 15. Two different location determination algorithms based on fingerprinting are then assessed in chapter 16. Finally, in chapter 17, the test results on location determination and navigation will be discussed. In the end, it is possible to answer the question:

How can Wi-Fi fingerprinting determine a location which does not utilize geometric features, and how can it be implemented in an application as such?

The conclusion is directly provided in part IV, as it further discusses the results related to it.

15. Surveying and databases

Android applications are composed by using at least two languages: Extensible Markup Language (XML) for the layout and interface, and Java for the functionality. On top of the Java, a separate Android library is installed, resulting in an adapted version of Java to Android. Additional libraries, such as an extended Wi-Fi library, can also be attached.

This application is no difference, and also utilizes the same style, as programmed in Eclipse (Figure 19). First, the interface was designed in XML. Screenshots are made via the emulator, which does not support Wi-Fi detection however (only the real device does so). The XML files can be portrayed at the client side of the application, with the server side being the inherited chain of Java files and databases, as there is no remote server for this prototype.

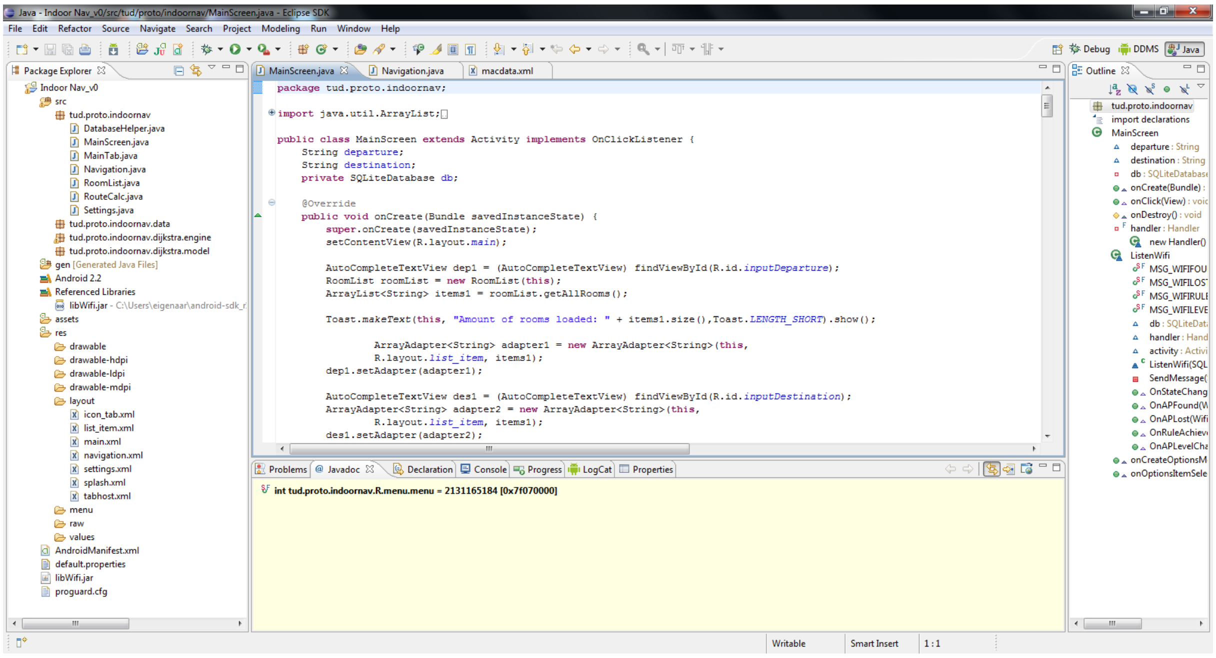


Figure 19. Screenshot of the Eclipse Environment

Chapter 15.1 first discusses the methodology of surveying and chapter 15.2 continues with the database implementation.

15.1. Site survey preparation

15.1.1. Measure points

The surveyed locations are parts of the ground, first and second floor of the OTB building at Delft University of Technology. In particularly, the OTB building wings 2 and 3, as wing 1 is reserved for another department of the university. As the office spaces in these wings can be locked, these are the only restricted areas in the test bed. In wing 3, at the ground, first and second floor, twelve points on each floor has been assigned a fingerprint, to ensure the fingerprints can be compared with each other regarding signal strengths, which allows a proper snapping: if the fingerprints would be too similar to each other, distinguishing the locations would be very hard (see also 16.2.3). Differences are obtained because of receiving direct and indirect signals due time, which are being caused by multipath effects. The points were measured in the aisles of the wings, at specific places, such as in front of the elevator, near the toilets, and on junctions of connecting office rooms. Figure

20 shows a photo of the location; Figure 22 displays the graphical representation of the measured points.

Figure 20. Photo of wing 2 and wing 3 of the OTB building

15.1.2. Measuring and determining suitable fingerprints

An ASUS K72J laptop, equipped with an Atheros AR9285 Wireless Network Adapter, has been used to register the signal strength in the first test runs. Signal strengths are expressed in a negative value of signal loss and noted with the unit dBm.

The signal strengths were measured with the inSSIDer 2.0 software, which could be downloaded at http://www.metageek.net/products/inssider/. inSSIDer 2.0 is able to register all received signal strengths, at a given interval, along with their unique MAC address. The uniqueness is necessary to distinguish access points with the same SSID, as otherwise measuring using SSIDs will result in collated signal strengths. At each location, a scan has been carried out for ten seconds, registering at a 1 second interval. Those locations were then tagged with location information, and merged into one file.

In the second test runs, new measurements have been performed, but instead of using a separate device to measure the fingerprints, the same device as the application is being used: the smartphone instead of the laptop has been used to do so. This is further elaborated in chapter 16.

The ten seconds scan, with a 1 second interval is necessary to determine stable, average signal strength, since the signal strengths can fluctuate in a short time, as displayed below in Figure 21: for instance at FID=201 (this has later been converted to FID=3201, to note the fingerprint is located in wing 3), from the first record: the signal strengths at this point changes between -43 dBm and -49 dBm, caused by the reception of both direct and indirect signals due multipath effects. It also detects the stability of access points: at a given location, if at least four out of ten are present, this can be considered as relatively stable. This will eliminate unstable access points, which are usually outlying routers. The information is first being stored in a SQL database from Microsoft Access. In Android, each database table can be stored in a separate XML file, using SQL statements instead of database tables. This conversion has been done as shown in annex A.4.

The next step is filtering the weak signals. It is possible that at a given location, a fingerprint consist of 30 MAC-addresses, in which the majority are below a RSSI of -80 dBm. It might be the case they will not appear in a scan at another given time. On the other hand, RSSIs higher than -70 dBm are most likely to be constantly present. Even then, it is possible that at a measured location the signal strength always remains relatively weak (for example, always lower than -70 dBm). In this process, it has been chosen that the 10 strongest signals picked up, will determine the position, or to be more precise: the fingerprint.

An unedited example is provided in Figure 21, ordered by location, then by average RSSI from highest to lowest. As can be seen, there are duplicate SSIDs, but there are different MACs. Also the number of detected signals within ten seconds changes. Do note that the location naming (FID) were preliminary in this figure. The final FIDs have been assigned as visible in Figure 22 (full graph) and Figure 23 (part of the graph on top of a photo).

MAC	SSID	RSSI_AVG	RSSI 1	RSSI 2	RSSI 3	RSSI 4	RSSI 5	RSSI 6	RSSI 7	RSSI 8	RSSI 9	RSSI 10	Location	FID
00:1A:A2:C0:0E:82	TUDelft-weblogin	**-46,0000**	-46	-45	-45	-45	-49	-49	-49	-43	-43		OTB-W3-F2-01020304	201
00:1A:A2:C0:0E:84	tudelft-dastud	**-46,1111**	-46	-45	-45	-45	-50	-50	-50	-42	-42		OTB-W3-F2-01020304	201
00:1A:A2:C0:0E:85	TUDelft-congress	**-46,1111**	-45	-45	-45	-45	-49	-49	-49	-44	-44		OTB-W3-F2-01020304	201
00:1A:A2:C0:0E:80	eduroam	**-46,9000**	-47	-47	-46	-47	-49	-49	-49	-47	-45	-43	OTB-W3-F2-01020304	201
00:1A:A2:FA:0F:A0	eduroam	**-71,0000**	-65	-65	-65	-65	-74	-74	-74	-76	-76	-76	OTB-W3-F2-01020304	201
00:1A:A2:FA:39:95	TUDelft-congress	**-71,2857**	-70	-70	-70	-70	-73	-73	-73				OTB-W3-F2-01020304	201
00:1A:A2:FA:0F:A4	tudelft-dastud	**-71,3000**	-65	-65	-65	-65	-75	-75	-75	-76	-76	-76	OTB-W3-F2-01020304	201
00:1A:A2:FA:0F:A5	TUDelft-congress	**-71,4000**	-66	-66	-66	-66	-75	-75	-75	-75	-75	-75	OTB-W3-F2-01020304	201
00:1A:A2:FA:0F:A2	TUDelft-weblogin	**-71,6000**	-65	-65	-65	-65	-76	-76	-76	-76	-76	-76	OTB-W3-F2-01020304	201
00:1A:A2:FA:39:94	tudelft-dastud	**-72,0000**	-75	-75	-75	-75	-69	-69	-69	-69	-72	-72	OTB-W3-F2-01020304	201
00:1A:A2:FA:39:90	eduroam	**-72,4000**	-75	-75	-75	-75	-70	-70	-70	-70	-72	-72	OTB-W3-F2-01020304	201
00:1A:A2:FA:39:92	TUDelft-weblogin	**-73,2000**	-77	-77	-77	-77	-69	-69	-69	-69	-74	-74	OTB-W3-F2-01020304	201
00:1A:A2:F9:F8:35	TUDelft-congress	**-79,2000**	-82	-82	-82	-82	-77	-77	-77	-77	-78	-78	OTB-W3-F2-01020304	201
00:1A:A2:FA:0E:E0	eduroam	**-80,0000**	-80	-80	-80	-80	-80	-80	-80	-80	-80	-80	OTB-W3-F2-01020304	201
00:1A:A2:FA:0E:E5	TUDelft-congress	**-81,0000**	-81	-81	-81	-81	-81	-81	-81	-81			OTB-W3-F2-01020304	201
00:1A:A2:FA:6C:F0	eduroam	**-86,0000**	-86	-86	-86	-86	-86	-86	-86	-86	-86	-86	OTB-W3-F2-01020304	201
00:1A:A2:C0:0E:80	eduroam	**-47,3000**	-49	-48	-47	-46	-49	-48	-47	-46	-47	-46	OTB-W3-F2-0506	202
00:1A:A2:C0:0E:82	TUDelft-weblogin	**-48,6667**	-48	-48	-48	-52	-52	-52	-46	-46	-46		OTB-W3-F2-0506	202
00:1A:A2:C0:0E:84	tudelft-dastud	**-49,0000**	-54	-54	-54	-45	-45	-45	-48	-48	-48		OTB-W3-F2-0506	202
00:1A:A2:C0:0E:85	TUDelft-congress	**-49,6667**	-52	-52	-52	-50	-50	-50	-47	-47	-47		OTB-W3-F2-0506	202
00:1A:A2:FA:0F:A2	TUDelft-weblogin	**-56,7000**	-57	-57	-57	-54	-54	-54	-54	-60	-60	-60	OTB-W3-F2-0506	202
00:1A:A2:FA:0F:A4	tudelft-dastud	**-57,0000**	-57	-57	-57	-54	-54	-54	-54	-61	-61	-61	OTB-W3-F2-0506	202
00:1A:A2:FA:0F:A5	TUDelft-congress	**-57,6000**	-58	-58	-58	-54	-54	-54	-54	-62	-62	-62	OTB-W3-F2-0506	202
00:1A:A2:FA:0F:A0	eduroam	**-58,0000**	-58	-58	-58	-55	-55	-55	-55	-62	-62	-62	OTB-W3-F2-0506	202
00:1A:A2:F9:F8:32	TUDelft-weblogin	**-67,1111**	-67	-67	-67	-67	-67	-67	-67	-67	-68		OTB-W3-F2-0506	202
00:1A:A2:F9:F8:34	tudelft-dastud	**-67,4444**	-68	-68	-68	-68	-67	-67	-67	-67	-67		OTB-W3-F2-0506	202
00:1A:A2:F9:F8:30	eduroam	**-68,0000**	-68	-68	-68	-68	-68	-68	-68	-68	-68		OTB-W3-F2-0506	202
00:1A:A2:F9:F8:35	TUDelft-congress	**-73,1000**	-74	-74	-74	-77	-77	-77	-77	-67	-67	-67	OTB-W3-F2-0506	202
00:1A:A2:FA:0E:E5	TUDelft-congress	**-76,1000**	-75	-75	-75	-77	-77	-77	-77	-76	-76	-76	OTB-W3-F2-0506	202
00:1A:A2:FA:0E:E0	eduroam	**-76,9000**	-80	-80	-80	-76	-76	-76	-76	-75	-75	-75	OTB-W3-F2-0506	202
00:1A:A2:FA:39:94	tudelft-dastud	**-79,2500**	-76	-76	-76	-81	-81	-81	-81	-82			OTB-W3-F2-0506	202
00:1A:A2:FA:39:90	eduroam	**-79,6250**	-76	-76	-76	-82	-82	-82	-82	-81			OTB-W3-F2-0506	202
00:1A:A2:FA:0E:E4	tudelft-dastud	**-79,8000**	-80	-80	-80	-80	-79						OTB-W3-F2-0506	202
00:1A:A2:FA:0E:E2	TUDelft-weblogin	**-80,0000**	-80	-80	-80	-80							OTB-W3-F2-0506	202
00:1A:A2:FA:39:95	TUDelft-congress	**-80,3333**	-80	-80	-80	-80	-80	-80	-80	-80	-83		OTB-W3-F2-0506	202
00:1A:A2:FA:39:92	TUDelft-weblogin	**-81,1000**	-76	-76	-76	-85	-85	-85	-85	-81	-81	-81	OTB-W3-F2-0506	202
00:23:5E:D6:04:00	CRESTRON	**-82,1429**	-56	-85	-85	-85	-88	-88	-88				OTB-W3-F2-0506	202
00:1A:A2:FA:6C:F0	eduroam	**-86,0000**	-86	-86	-86	-86	-86	-86	-86	-86	-86	-86	OTB-W3-F2-0506	202

Figure 21. Screenshot of reconfigured RSSI scan

In this figure, temporarily constructed Location and FID tags have been generated. The values RSSI1-RSSI10 represents measurements within ten seconds, where the number of RSSI depicts the second.

As displayed in the Figure 21, within ten seconds, using an interval of one second, RSSIs from several routers (MAC+SSID) at specific locations (Location/FID) are measured and recorded. These results in a maximum of ten values (RSSI1-RSSI10), from these values an average can be calculated (RSSI_AVG).

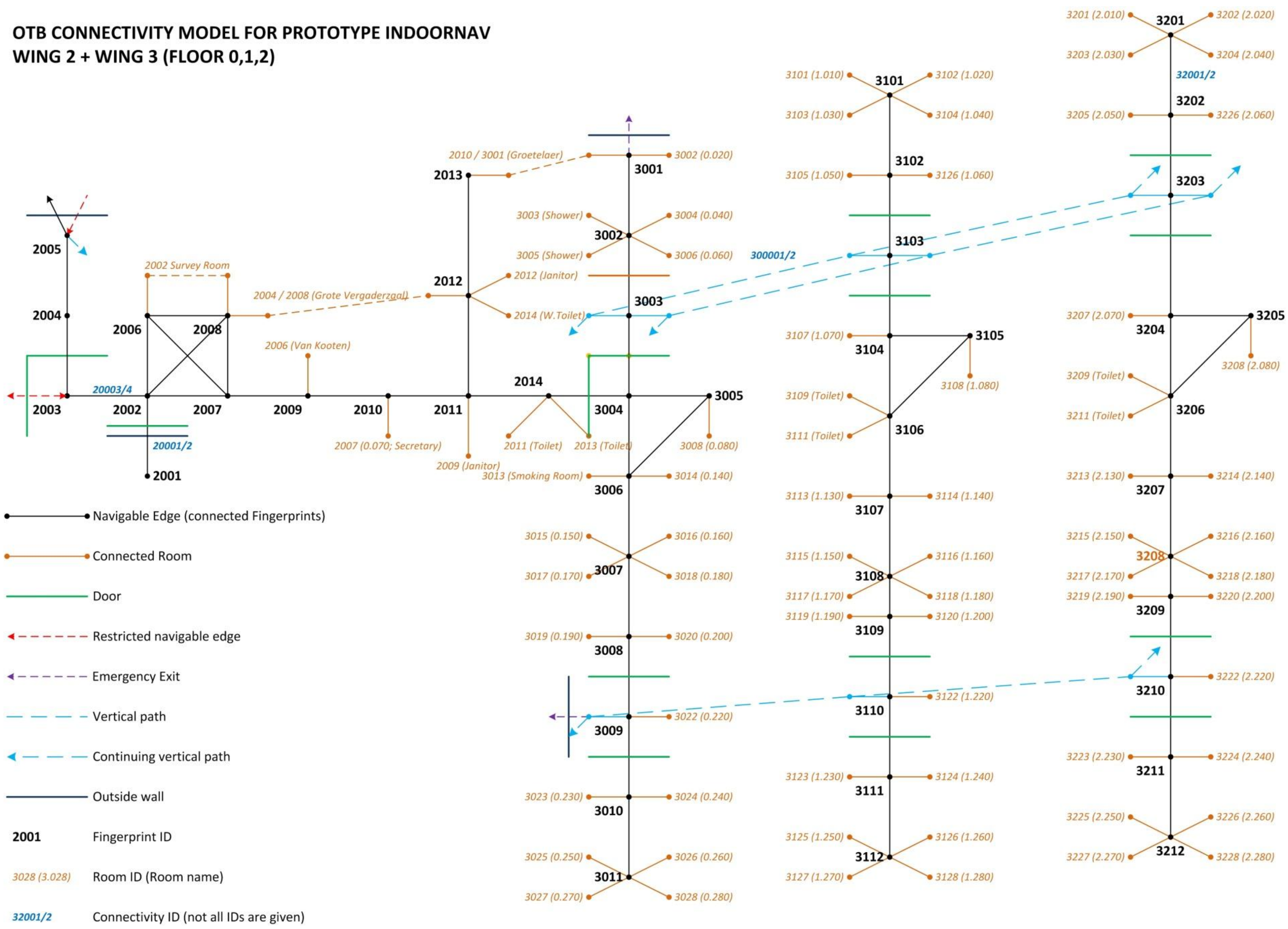


Figure 22. Graphical representation of indoor model (graph) with correct orientation

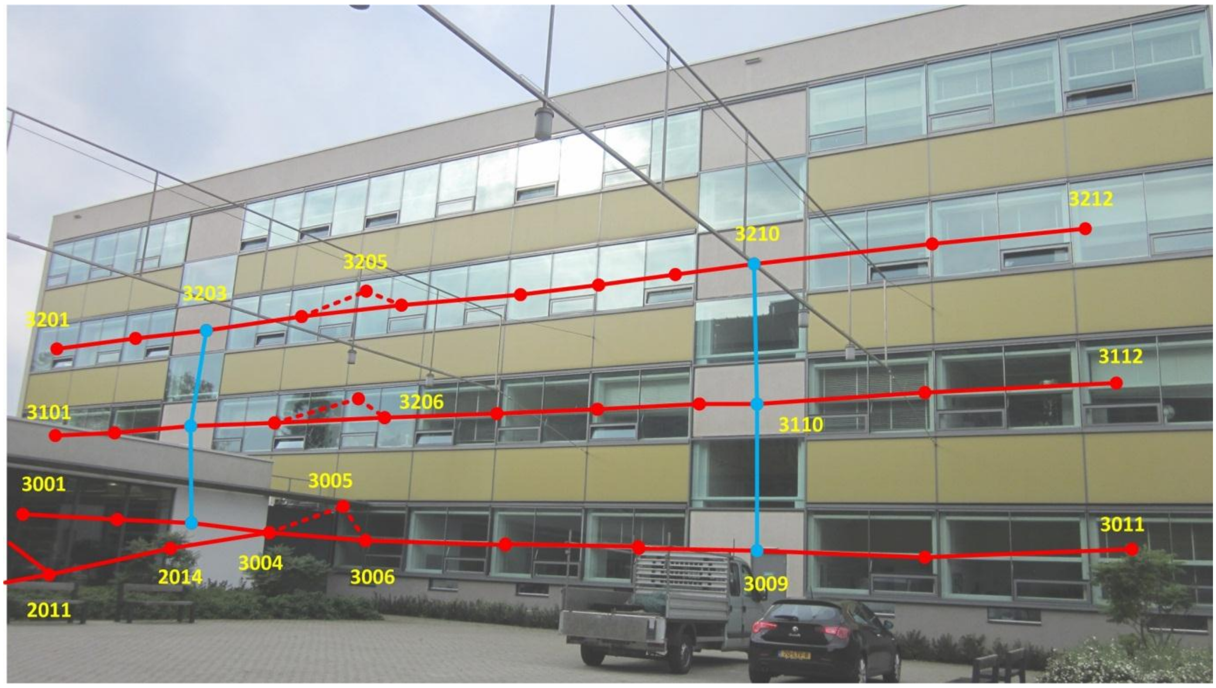


Figure 23. Graph layer on photo

15.1.3. Fingerprinting methodology used in this project

Because the Wi-Fi signal carrier is not consistent itself, it is hard to identify a location without a fixed signal strength. Signal strengths vary over time, caused by multipath, and these signal strengths themselves are not consistent on top of that. However, by averaging the signal strength values, and by creating a search space (a range) around this average, it is still possible to predict a location. This is the methodology being used in this project. This is illustrated in Figure 24: a recorded fingerprint deviates in signal strengths when it is compared at a later time stamp. At the x-axis, the signal strengths are plotted for one fingerprint. Other fingerprints are being measured simultaneously, which can be plotted at the z-axis in the figure below.

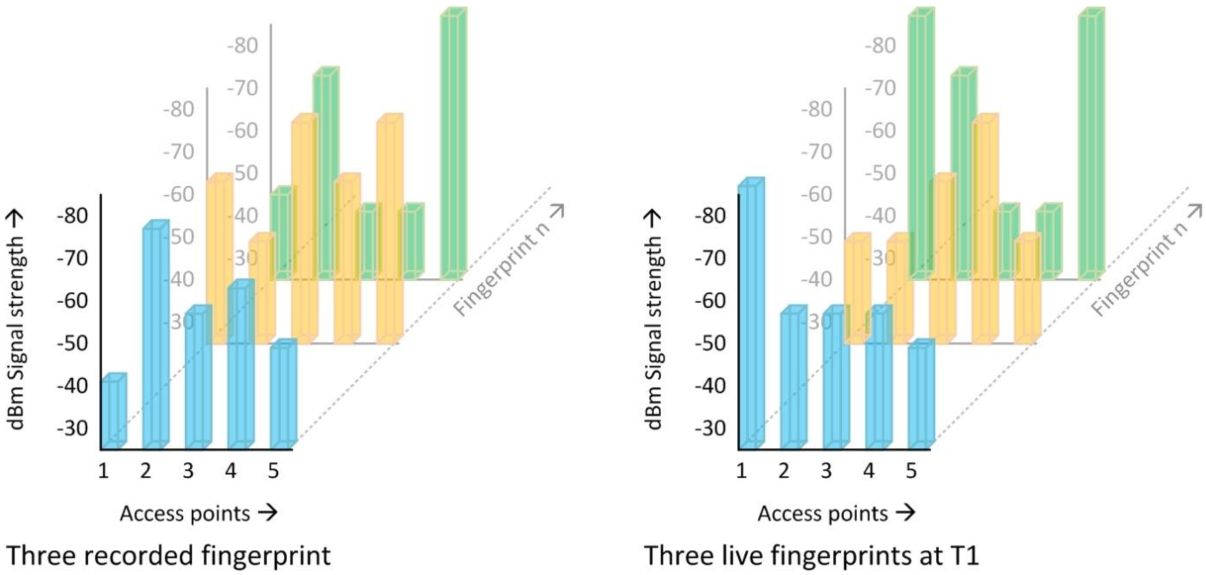


Figure 24. Time aspect of fingerprinting

At time stamp 1, at a specific location, a certain amount of signal strength is being received, which has been obtained directly. At time stamp 2, at the same location, the amount has been changed, because it received the signal strength delayed because of the multipath effect. For example, in Figure 25/Table 6, at location 3210, there is a given signal from MAC 00:1A:A2:FA:0E:B0 of -72 dBm at time stamp T1, which is received directly. At time stamp T3, it received a delayed signal strength caused by multipath of -78 dBm, after a while, it receives another value of -74 dBm at T7. To counter these differences, an average has been taken based on 10 time stamps.

MAC	00:1A:A2:FA:0E:B0
SSID	eduroam
FID	3210
AVG	-75,2
T1	-72
T2	-72
T3	-78
T4	-78
T5	-78
T6	-78
T7	-74
T8	-74
T9	-74
T10	-74

Table 6. Temporal aspect of fingerprinting (table view)

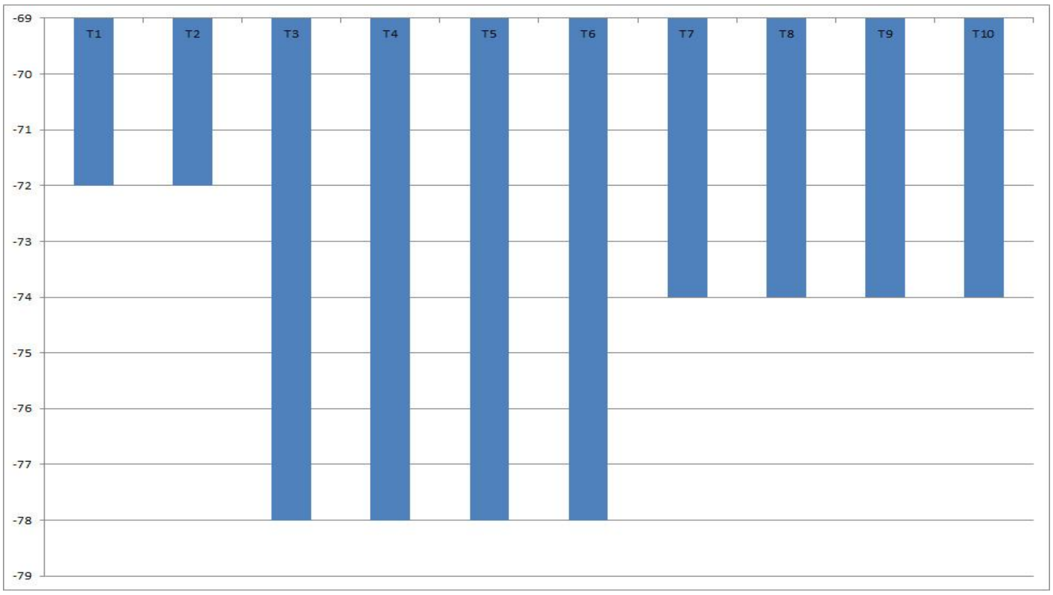


Figure 25. Temporal aspect of fingerprinting (bar chart) for given location FID=3210

Values in both Figure 25/Table 6 are in dBm.

15.2. Databases and information storage

15.2.1. Generalities

In this prototype, the database is stored in the application, as Android allows doing so with SQLite (Android Developers, 2011). If a large scale service deployment could be realized, a large database can ideally be stored on a server, in which the user needs a connection to obtain information. Since the database of the prototype is rather small, it is not necessary to work with a remote server. This results in a standalone mobile application.

Originally, several database tables have been proposed, which included information about location (fingerprints and general separated), connectivity, location related information (points of interests and room-specific information), and personal information. However, personal information is heavily restricted due to privacy and legal issues, in combination with locational information. In the final prototype, the personal information has thus been left out. This issue is further discussed in chapter 18. Moreover, the table_poi has been merged with the table_rooms, as ideally, POIs have to be assigned with the same location FIDs. This decreases probabilities in data redundancy as well.

Another prototype restriction is the availability of selected buildings. In a large scale service deployment, the user is able to browse through various buildings, at given addresses and places. Since the prototype only contains one building, the option to search for locations in other buildings than the OTB has been disabled. More discussion is to be found in chapter 19 (limitations)

In Figure 26, an overview of these relations and database tables can be seen, along with their data type written in capitals. The database fields are further explained in Annex A.3, with the exception of the table for persons, since that has been omitted (see chapter 20 on discussion on privacy issues). Table_mac has become an independent table, and only shows the MAC address along with their RSSI, at given locations, which is joined by the original location table.

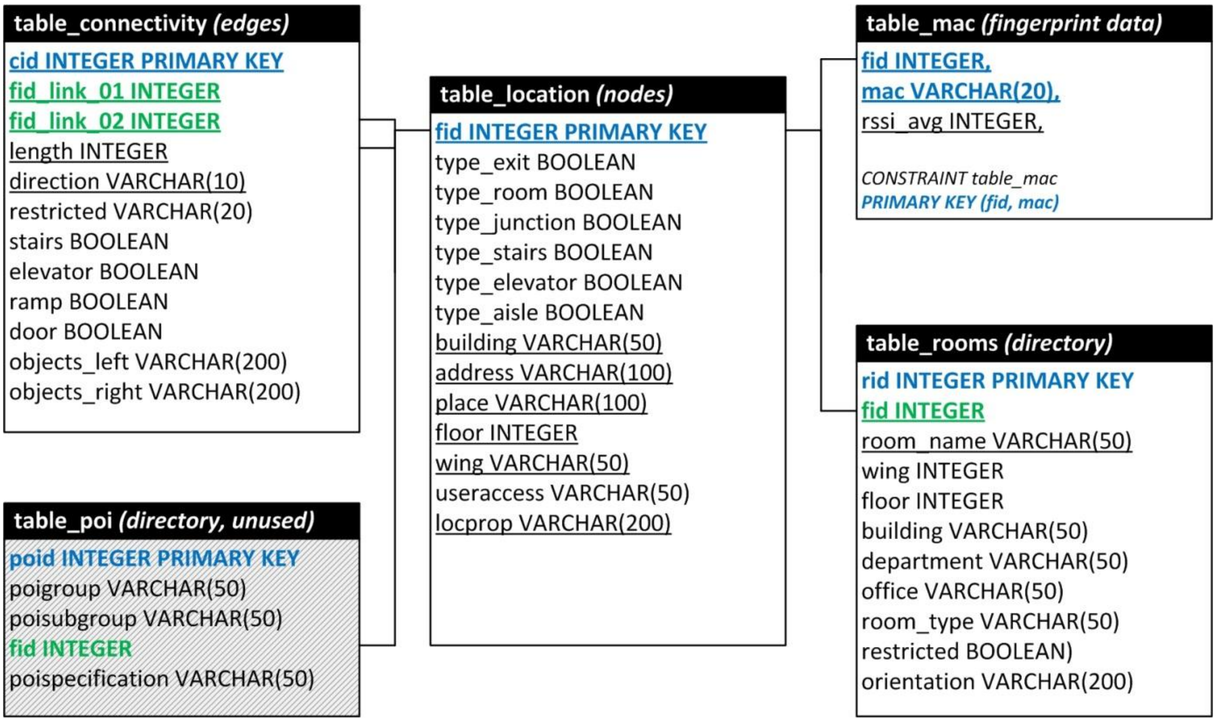


Figure 26. Database relations

The most important fields include the location ID from the table_location, which is used in any other table. This coincides with the fingerprint, as the fingerprint is the node itself. The table_location states the address, location, and type of location (whether this is an exit, a room or stairs as such). The table_connectivity provides information on the connection between the fingerprints; each link or edge has a unique ID (the connection ID, or the cid), and states whether it traverses over objects, as well as which objects are to the left or to the right. In the overview above, the fields which are underlined are truly used in the prototype. This effectively means that any double field in other database tables can be removed, decreasing the data redundancy.

15.2.2. The database in the application

The database has been inserted as SQL statements in separate XML files, as databases in Android can be configured as such. In Box 3 below, an example of a XML file is shown.

```
<sql>
        <statement>
                CREATE TABLE IF NOT EXISTS table_rooms (
                rid INTEGER PRIMARY
                KEY,
                fid INTEGER,
                room_name VARCHAR(50),
                wing INTEGER,
                floor INTEGER,
                building VARCHAR(50),
                department VARCHAR(50),
                office VARCHAR(50),
                room_type VARCHAR(50),
                restricted BOOLEAN))
</statement>
        <statement>INSERT INTO table_rooms VALUES(2002,2008,'Survey
                Room',2,0,'OTB','','Room','Computer Room',1);</statement>
        <statement>INSERT INTO table_rooms VALUES(2004,2008,'Grote
                Vergaderzaal',2,0,'OTB','','Room','Commission',1);</statement>
        <statement>INSERT INTO table_rooms VALUES(2006,2009,'Anneke van
                Cootenzaal',2,0,'OTB','','Room','Commission',1);</statement>
        <statement>INSERT INTO table_rooms VALUES(2007,2010,'0.070
                (Secretary)',2,0,'OTB','','Work Office','Office',1);</statement>
```

Box 3. Example of SQL statements in XML: creation and population of database table

As such, an XML document can be filled with statements, in which those statements can be read in the Java file again. Other examples are included in the annex A.4.

The database can be read with the use of a Java class “DatabaseHelper.java” which utilizes the XML/SQL statements. It is up to the programmer to define which fields can be used for which purpose. The fields that the user gets to see, are the room names, and the object information, which is stored in the table_connectivity (object_left and object_right) and table_location (loc_prop). The room name is coupled with a room id (rid), which is combined with a specific location (fid). The user gets to see a reverse geocoded specific location: a specific fingerprint (mac+rssi from the table_mac), can be found, which is then looked up in the database table, which room and location belongs to that fingerprint.

In Box 4 below the database is called from the DatabaseHelper.java file. In this file, the `parseXmlFile(R.raw.roomsdata, db);` states that the XML file, which contains SQL statements, roomsdata is to be used as a database. This coincides with the contents of the table_rooms as shown in Box 3. The statement `NodeList statements = doc.getElementsByTagName("statement");` explicitly tells that the SQL statements can be found in the XML document, which has the tags <statement>

```
public class DatabaseHelper extends SQLiteOpenHelper {

        public static final String DATABASE_NAME = "rooms";

        protected Context context;

        public DatabaseHelper(Context context) {
                super(context, DATABASE_NAME, null, 1);
                this.context = context;
        }

        @Override
        public void onCreate(SQLiteDatabase db) {
                parseXmlFile(R.raw.roomsdata, db);
                parseXmlFile(R.raw.connectiondata, db);
                parseXmlFile(R.raw.macdata, db);
                parseXmlFile(R.raw.poidata, db);
                parseXmlFile(R.raw.locationdata, db);
        }

        private void parseXmlFile(int id, SQLiteDatabase db) {
                String s;
                Log.i("DbHelper", "Executing SQL statements");

                try {
//                      Toast.makeText(context, "1", 2000).show();
                        InputStream in = context.getResources().openRawResource(id);
                        DocumentBuilder builder =
DocumentBuilderFactory.newInstance().newDocumentBuilder();
                        Document doc = builder.parse(in, null);
                        NodeList statements = doc.getElementsByTagName("statement");
                        for (int i=0; i<statements.getLength(); i++) {
                                s = statements.item(i).getChildNodes().item(0).getNodeValue();
                                db.execSQL(s);
                        }
                } catch (Throwable t) {
                        Toast.makeText(context, t.toString(), 50000).show();
                        Log.e("DbHelper", t.getMessage());
                }
        }

        @Override
        public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
                db.execSQL("DROP TABLE IF EXISTS table_rooms");
                onCreate(db);
        }

}
```

Box 4. Snippet overview of DatabaseHelper.java: getting the contents from database tables from XML files

The database can be further used for input for the client. In the main screen, the user can select and enter rooms and locations from the table_rooms to calculate a route. The MainScreen.java is shown in Box 5. The `setContentView(R.layout.main);` refers to the fact that the main.xml document has to be used as layout. Then, there are two fields which need to be connected to the database, in where the client has to provide information on where the client wants to go, and what the client's departure place is. This can be done by using an AutoCompleteTextView, which is an extension of Android, where the user have to type something, which will be automatically completed by a list, so that the user can select an item from the list. This is being recalled with:

```
AutoCompleteTextView dep1 = (AutoCompleteTextView) findViewById(R.id.inputDeparture);
              RoomList roomList = new RoomList(this);
              ArrayList<String> items1 = roomList.getAllRooms();
```

However, it must explicitly state which fields are required to display. Then, a separate Java file needs to be written, which is shown in Box 6.

Box 5. Snippet overview of MainScreen.java: creation of entry fields and actions upon tapping buttons

```
public class RoomList {
        protected SQLiteDatabase db;
        protected Cursor cursor;

        public RoomList(Context c) {
                Log.i("Roomlist", "Creating DB");

                db = (new DatabaseHelper(c).getWritableDatabase());
        }

        public ArrayList<String> getAllRooms() {
                // Log.i("Roomlist", "Query execute");

                Cursor cursor = db.rawQuery("SELECT rid, room_name FROM table_rooms",
                                null);
                ArrayList<String> rooms = new ArrayList<String>();

                // Log.i("Roomlist", cursor.getCount()+"Amount found");

                if (cursor.getCount() >= 1) {
                        while (cursor.moveToNext()) {
                                rooms.add(cursor.getString(1));

                        }
                }
                cursor.close();
                return rooms;
        }
}
```

Box 6. Snippet overview of RoomList.java: populating an entry field with a directory service

A query must be formed first, in order to populate the AutoCompleteTextView fields. First this field (an Array List) must be formed, which is done by `public ArrayList<String> getAllRooms()`, next the query will be summoned and filled in the Array List, with:

```
Cursor cursor = db.rawQuery("SELECT rid, room_name FROM table_rooms", null);
                ArrayList<String> rooms = new ArrayList<String>();
```

Then this step will be repeated for the entire database table, which is done by:

```
if (cursor.getCount() >= 1) {
                        while (cursor.moveToNext()) {
                                rooms.add(cursor.getString(1));

                        }
                }
                cursor.close();
                return rooms;
```

This way, the AutoCompleteTextView fields are fully populated. Both departure and destination can use the same Array List. A message will be briefly shown, how many rooms are added:

```
Toast.makeText(this, "Amount of rooms loaded: " + items1.size(),
                                Toast.LENGTH_SHORT).show();
```

However, upon confirming the fields, one must be certain that there are no empty fields, or that the same records are being used for departure as destination, as soon as the client clicked on the button to calculate a route. This is solved with:

```
public void onClick(View v) {
        if (v.getId() == R.id.buttonNavigationStart) {
        TextView inputDeparture = (TextView) findViewById(R.id.inputDeparture);
        TextView inputDestination = (TextView) findViewById(R.id.inputDestination);

                if (inputDeparture.getText().length() == 0) {
                        Toast.makeText(this, "Please input departure",
                                        Toast.LENGTH_SHORT).show();
                        return;
                } else if (inputDestination.getText().length() == 0) {
                        Toast.makeText(this, "Please input destination",
                                        Toast.LENGTH_SHORT).show();
                        return;

                } else if (inputDeparture == inputDestination) {
                        Toast.makeText(this, "Departure and Destination is the same",
                                        Toast.LENGTH_SHORT).show();
                        return;
                }

                if (false) {
                }
                Intent n = new Intent(this, Navigation.class);
                startActivity(n);
        } else if (v.getId() == R.id.buttonSettings) {
                Intent s = new Intent(this, Settings.class);
                startActivity(s);
```

As the following Java codes checks if the field is empty:

```
if (inputDeparture.getText().length() == 0) {
                        Toast.makeText(this, "Please input departure",
                                        Toast.LENGTH_SHORT).show();
                        return;
```

In this prototype, the decision has been made not to combine the current location with an input field for origin point, to allow the user to calculate a route in advance, without being on the present spot. This might be useful if the user would like to prepare his route, rather than finding out on the fly.

16. Location determination algorithms

The prototype resembles three major functionalities:

1. The determination of a location;
2. Routing between pre-determined locations or current location with destination;
3. Route confirmation during the navigation.

The routing itself is pre-determined in the sense that the user is able to type and select specific locations, but the application only accepts predefined locations. This is comparable with planning railway journeys, where there are only a limited number of stations to travel between. The user can only use a railway station from a database. In the prototype, the user can only select rooms or POIs and the user does not necessarily need the sensor to calculate a route. This can be useful if the user is planning ahead. The route confirmation information is then not usable.

From here, it is desirable to combine two major functionalities in a fourth aim:

4. To navigate between locations, where the current location is constantly adapted to the users situation, as the user sees the current location constantly at the top half of the screen.

Section 16.1 describes which types of location determination is being used in this thesis. Section 16.2 continues with the translation of the obtained locations, while section 16.3 continues with how the previous sections were programmed. Finally, section 16.4 deals with the aspect of navigation and route confirmation.

16.1. Two methods of location determination being used in this thesis

Method 1: count method with search space

One possible method to determine the location, is by picking up the signal strengths in negative dBm per MAC address. These values have to be compared to a recorded database, in which location information can be returned. This results in the following algorithm:

1. Pick up signal strength on the fly (within 1 second).
2. Snap signals within a range of *n* dBm.
3. Compare the live received signal strengths with the signal strengths ranges in the database (which is a range query). Count the number of matches per FID.
4. Return the location ID of the highest match. Based on the location ID, return additional information from other tables as well.
5. If there are multiple matches, take both.
6. To prepare and speed up the prediction of the next location, eliminate outlying locations (see also chapter 14.3).

The databases that will be used are displayed below.

First, the existing databases on signal strengths:

FID	MAC	RSSI
3001	00:1a:a2:c0:0e:85	-74
3001	00:1a:a2:c0:0e:82	-75
3001	00:1a:a2:fa:0f:a0	-59
3001	00:1a:a2:fa:0f:a2	-60
3001	00:1a:a2:fa:0f:a4	-60
3001	00:1a:a2:fa:0f:a5	-59
3001	00:1a:a2:fa:39:90	-41
3001	00:1a:a2:fa:39:92	-49
3001	00:1a:a2:fa:39:94	-49
3001	00:1a:a2:fa:39:95	-48

FID	MAC	RSSI
3002	00:1a:a2:fa:0f:a0	-63
3002	00:1a:a2:fa:0f:a2	-62
3002	00:1a:a2:fa:0f:a4	-64
3002	00:1a:a2:fa:0f:a5	-64
3002	00:1a:a2:fa:25:40	-72
3002	00:1a:a2:fa:25:45	-72
3002	00:1a:a2:fa:39:90	-42
3002	00:1a:a2:fa:39:92	-43
3002	00:1a:a2:fa:39:94	-43
3002	00:1a:a2:fa:39:95	-43

Table 7. Sample database on signal strengths on two different locations

The following information is available about the location itself (the table has been truncated in comparison to the actual table for this purpose):

FID	Building	Address	Place	Floor	Wing	Access	Locprop_01
3001	OTB	Jaffalaan 9	Delft	0	3	0	Emergency Exit
3002	OTB	Jaffalaan 9	Delft	0	3	0	Showers

Table 8. Sample database on location information

The following information is available about the rooms (the table has been truncated in comparison to the actual table for this purpose):

RID	FID	Room_name	Wing	Floor	Office	Room_Type	Restricted
3001	**3001**	Room 0.010 (Pieter Groetelaerszaal)	3	0	Commission Room	Room	1
3002	**3001**	Room 0.020	3	0	Work Office	Office	1
3003	**3001**	Room 0.030 (Shower)	3	0	Shower	Shower	0
3004	**3002**	Room 0.040	3	0	Work Office	Office	1
3005	**3002**	Room 0.050 (Shower)	3	0	Shower	Shower	0
3006	**3002**	Room 0.060	3	0	Work Office	Office	1

Table 9. Sample database on room information

Note that one node (FID) contains several navigable spaces (RID), as the level of detail allows this. If it is desirable to make the rooms separate navigable as well (for instance, two specific points of interests are located inside a room), then separate fingerprints and thus nodes will need to be added to the database.

Now, step by step the algorithm will be explained.

Step 1: pick up signal strength

During the receiving phase, the following signals are retrieved, as displayed in Table 10:

MAC	RSSI
00:1a:a2:c0:0e:85	-78
00:1a:a2:c0:0e:82	-78
00:1a:a2:fa:0f:a0	-61
00:1a:a2:fa:0f:a2	-61
00:1a:a2:fa:0f:a4	-61
00:1a:a2:fa:0f:a5	-61
00:1a:a2:fa:25:40	-75
00:1a:a2:fa:25:45	-75
00:1a:a2:fa:39:90	-43
00:1a:a2:fa:39:92	-50
00:1a:a2:fa:39:94	-50
00:1a:a2:fa:39:95	-49

Table 10. Example of received signal strengths

Step 2: create range for the signal strength with ±n dBm

After filtering the weak and unstable RSSIs, an average RSSI can be calculated. However, since the RSSIs fluctuate over time and space, a snapping tolerance is necessary to determine a position. Therefore, a standard deviation has been calculated over all measurements, which was an average of 2.5 dBm (without filtering, this was 2.4 dBm). As a result, a rounding off to 3.0 dBm has been accepted as the snapping tolerance. To make sure at least a signal was being recognized, an additional 1.0 dBm has been added, resulting in a 4 dBm search space margin. If the received signal strength in the online phase is within a range of 4 dBm, then the measured signal matches the location recorded signal. This is illustrated in Figure 27.

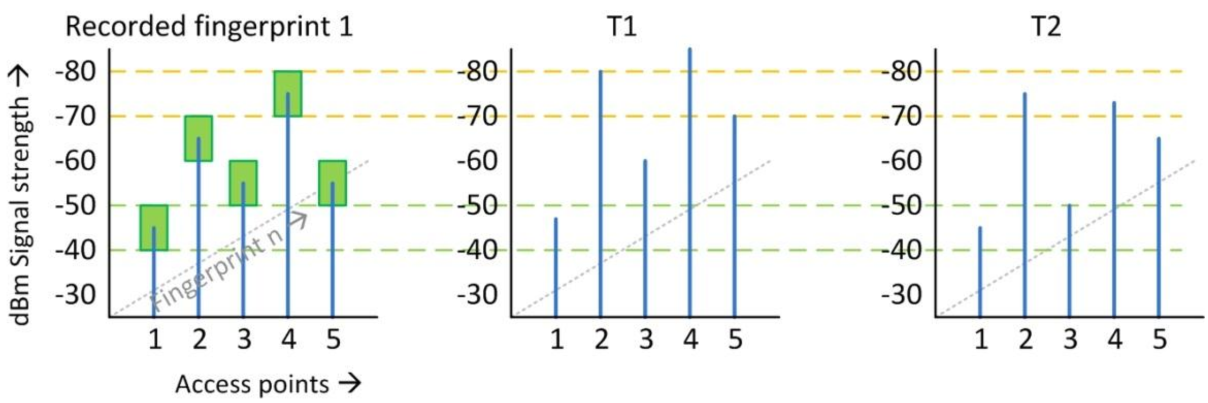


Figure 27. Matching live fingerprints with recorded fingerprints using snapping tolerances

The values in this figure are purely indicative and do not represent actual values, nor snapping values

From the received signal strengths, *n* dBm has to be added and subtracted to get a range. For example, if a snapping value of 3 dBm has been used, the result is the following Table 11:

MAC	Lower value	Upper value
00:1a:a2:c0:0e:85	-75	-81
00:1a:a2:c0:0e:82	-75	-81
00:1a:a2:fa:0f:a0	-58	-64
00:1a:a2:fa:0f:a2	-58	-64
00:1a:a2:fa:0f:a4	-58	-64
00:1a:a2:fa:0f:a5	-58	-64
00:1a:a2:fa:25:40	-72	-78
00:1a:a2:fa:25:45	-72	-78
00:1a:a2:fa:39:90	-40	-46
00:1a:a2:fa:39:92	-47	-53
00:1a:a2:fa:39:94	-47	-53
00:1a:a2:fa:39:95	-46	-52

Table 11. Example of determination of range signal strengths

Step 3: comparison and matching the database

This step exists of four sub steps:

A. From the existing database table_mac, compare if the recorded value is within the range of received signal strengths table.
B. If this is being the case, accept the value. If this is not being the case, discard the value.
C. Count the number of matches per fingerprint FID; accept the FID from the recorded database.
D. If multiple IDs were matching, then probably the user finds itself at a location between two recorded places. Both IDs can be returned. If no FIDs were matching, the statement “Location not found” can be returned.

First, the received signal strengths are now being compared with the recorded signal strengths and the signals are being rejected or accepted (step A and B, Table 12):

MAC	Received	Lower	Upper	FID=3001	Accepted	FID=3002	Accepted
00:1a:a2:c0:0e:85	-78	-75	-81	-74	NO		
00:1a:a2:c0:0e:82	-78	-75	-81	-75	YES		
00:1a:a2:fa:0f:a0	-61	-58	-64	-59	YES	-63	YES
00:1a:a2:fa:0f:a2	-61	-58	-64	-60	YES	-62	YES
00:1a:a2:fa:0f:a4	-61	-58	-64	-60	YES	-64	YES
00:1a:a2:fa:0f:a5	-61	-58	-64	-59	YES	-64	YES
00:1a:a2:fa:25:40	-75	-72	-78			-72	YES
00:1a:a2:fa:25:45	-75	-72	-78			-72	YES
00:1a:a2:fa:39:90	-43	-40	-46	-41	YES	-42	YES
00:1a:a2:fa:39:92	-50	-47	-53	-49	YES	-43	NO
00:1a:a2:fa:39:94	-50	-47	-53	-49	YES	-43	NO
00:1a:a2:fa:39:95	-49	-46	-52	-48	YES	-43	NO

Table 12. Example of comparing recorded and received signal strengths

Then, the number of matches needs to be counted. The result is 9 and 7 out of 10, respectively (step C and D). FID=3001 is being accepted as the location, FID=3002 is being rejected. The user finds itself near FID=3001.

Step 4: returning the information

The user has no information to extract from solely “FID=3001”. As soon as the matching took place, information has to be returned. If the user is located near FID=3001, information from Table 8 and Table 9 can be returned, such as: “*you are near the emergency exit*”, or “*you are near Room 0.010, Room 0.020 and Room 0.030*”, at “*Wing 3, Floor 0*”.

The location information can also be used to monitor the actual location and from there recalculate a route, if necessary.

The procedure steps 1-4 have been visualized in a flow scheme, as visible in Figure 28. Figure 29 shows a visualization of the process upon walking from one end to another: as the number of matches are being counted.

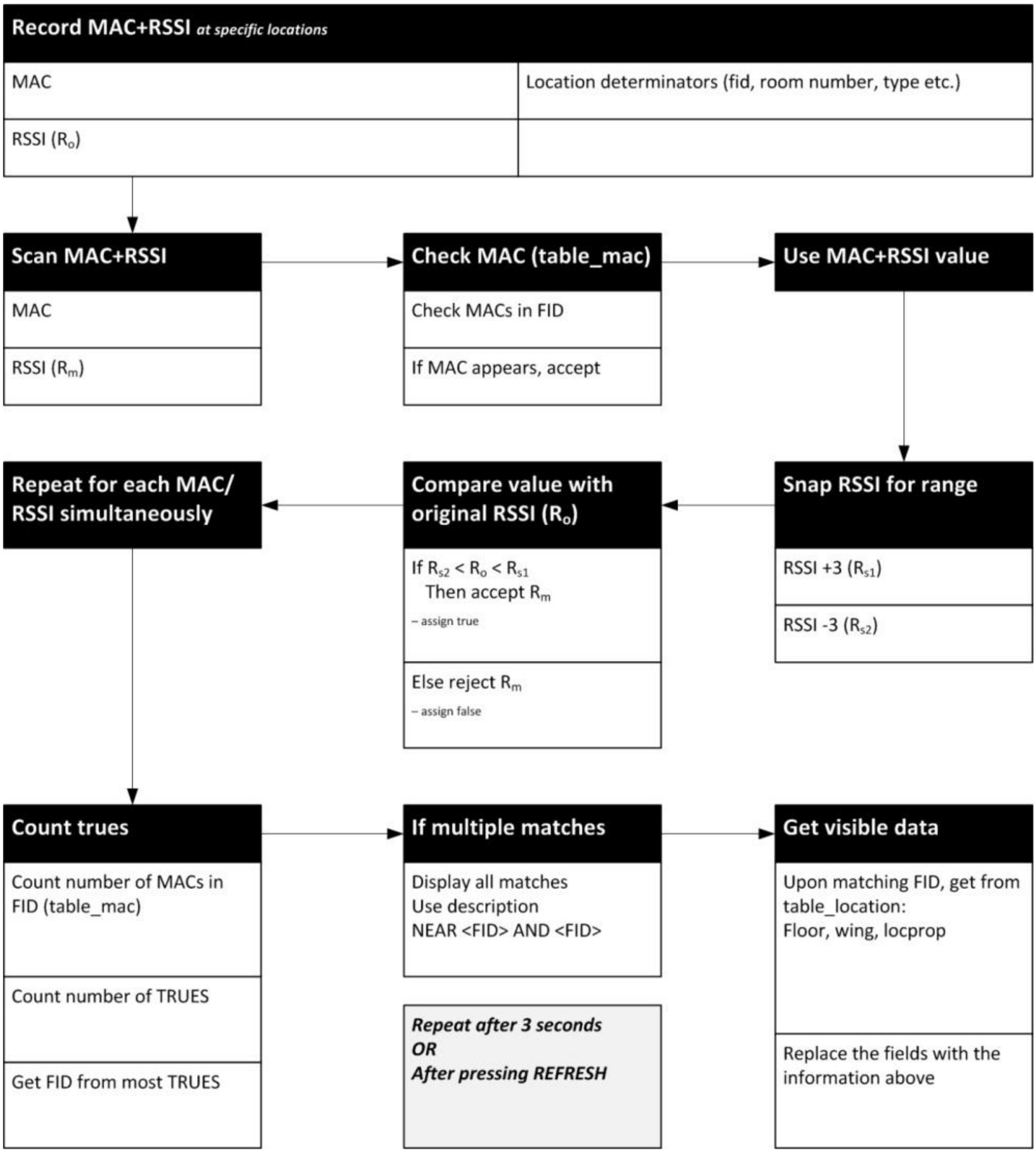


Figure 28. Flow scheme of location determination for count method with search space

Figure 29 is purely illustrative (it is not to scale and not linear) and simulates how the detection and counting of correct fingerprints should take place upon walking around.

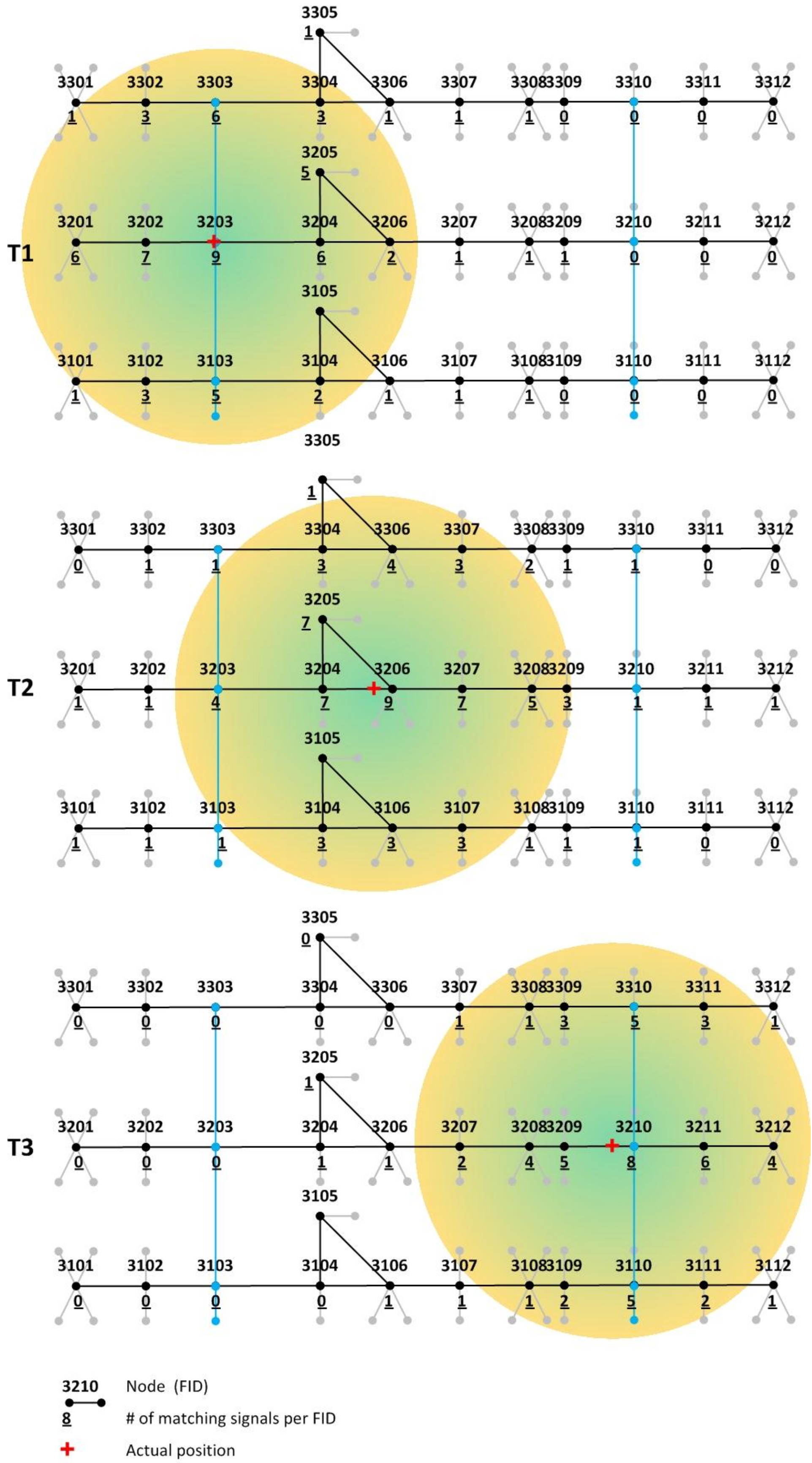


Figure 29. Visual representation of count method upon walking around (desirable simulation)

Method 2: Least sum of square method for location determination

It is also possible to make use of a least sum of square (SS) method as proposed by Shum et al. (2011). Instead of counting the number of matches, the differences between the actual and surveyed are being used, similar to taking a physical distance into account. The lower the SS value, the most likely it is the user finds itself at the predicted location. An example has been showed in Table 13 below – to the far right the outcome of the SS method can be seen.

		RSSI		Range 3			Range 5			Range 10			SS Method		
FID	MAC	O	M	L	H	1	L	H	1	L	H	1	DIF	DIF²	SS
3001	00:1A:A2:C0:0E:82	-75	-78	-78	-72	1	-80	-73	1	-85	-65	1	3	9	107
3001	00:1A:A2:C0:0E:85	-74	-78	-77	-71	0	-79	-72	1	-84	-64	1	4	16	
3001	00:1A:A2:FA:0F:A0	-59	-63	-62	-56	0	-64	-57	1	-69	-49	1	4	16	
3001	00:1A:A2:FA:0F:A2	-60	-62	-63	-57	1	-65	-58	1	-70	-50	1	2	4	
3001	00:1A:A2:FA:0F:A4	-60	-62	-63	-57	1	-65	-58	1	-70	-50	1	2	4	
3001	00:1A:A2:FA:0F:A5	-59	-63	-62	-56	0	-64	-57	1	-69	-49	1	4	16	
3001	00:1A:A2:FA:39:90	-41	-45	-44	-38	0	-46	-39	1	-51	-31	1	4	16	
3001	00:1A:A2:FA:39:92	-49	-45	-52	-46	0	-54	-47	0	-59	-39	1	-4	16	
3001	00:1A:A2:FA:39:94	-49	-46	-52	-46	1	-54	-47	0	-59	-39	1	-3	9	
3001	00:1A:A2:FA:39:95	-48	-47	-51	-45	1	-53	-46	1	-58	-38	1	-1	1	
3002	00:1A:A2:FA:0F:A0	-63	-63	-66	-60	1	-68	-61	1	-73	-53	1	0	0	77
3002	00:1A:A2:FA:0F:A2	-62	-62	-65	-59	1	-67	-60	1	-72	-52	1	0	0	
3002	00:1A:A2:FA:0F:A4	-64	-62	-67	-61	1	-69	-62	1	-74	-54	1	-2	4	
3002	00:1A:A2:FA:0F:A5	-64	-63	-67	-61	1	-69	-62	1	-74	-54	1	-1	1	
3002	00:1A:A2:FA:25:40	-72	-77	-75	-69	0	-77	-70	1	-82	-62	1	5	25	
3002	00:1A:A2:FA:25:45	-72	-75	-75	-69	1	-77	-70	1	-82	-62	1	3	9	
3002	00:1A:A2:FA:39:90	-42	-45	-45	-39	1	-47	-40	1	-52	-32	1	3	9	
3002	00:1A:A2:FA:39:92	-43	-45	-46	-40	1	-48	-41	1	-53	-33	1	2	4	
3002	00:1A:A2:FA:39:94	-43	-46	-46	-40	1	-48	-41	1	-53	-33	1	3	9	
3002	00:1A:A2:FA:39:95	-43	-47	-46	-40	0	-48	-41	1	-53	-33	1	4	16	
3003	00:1A:A2:FA:0F:A0	-61	-63	-64	-58	1	-66	-59	1	-71	-51	1	2	4	1013
3003	00:1A:A2:FA:0F:A2	-60	-62	-63	-57	1	-65	-58	1	-70	-50	1	2	4	
3003	00:1A:A2:FA:0F:A4	-61	-62	-64	-58	1	-66	-59	1	-71	-51	1	1	1	
3003	00:1A:A2:FA:0F:A5	-61	-63	-64	-58	1	-66	-59	1	-71	-51	1	2	4	
3003	00:1A:A2:FA:25:42	-63	-62	-66	-60	1	-68	-61	1	-73	-53	1	-1	1	
3003	00:1A:A2:FA:25:44	-62	-62	-65	-59	1	-67	-60	1	-72	-52	1	0	0	
3003	00:1A:A2:FA:39:90	-62	-45	-65	-59	0	-67	-60	0	-72	-52	0	-17	289	
3003	00:1A:A2:FA:39:92	-62	-45	-65	-59	0	-67	-60	0	-72	-52	0	-17	289	
3003	00:1A:A2:FA:39:94	-61	-46	-64	-58	0	-66	-59	0	-71	-51	0	-15	225	
3003	00:1A:A2:FA:39:95	-61	-47	-64	-58	0	-66	-59	0	-71	-51	0	-14	196	
3004	00:1A:A2:FA:0F:A0	-64	-63	-67	-61	1	-69	-62	1	-74	-54	1	-1	1	1628
3004	00:1A:A2:FA:0F:A4	-63	-62	-66	-60	1	-68	-61	1	-73	-53	1	-1	1	
3004	00:1A:A2:FA:0F:A5	-64	-63	-67	-61	1	-69	-62	1	-74	-54	1	-1	1	
3004	00:1A:A2:FA:25:40	-59	-77	-62	-56	0	-64	-57	0	-69	-49	0	18	324	
3004	00:1A:A2:FA:25:42	-57	-62	-60	-54	0	-62	-55	1	-67	-47	1	5	25	
3004	00:1A:A2:FA:25:44	-59	-62	-62	-56	1	-64	-57	1	-69	-49	1	3	9	
3004	00:1A:A2:FA:25:45	-59	-75	-62	-56	0	-64	-57	0	-69	-49	0	16	256	
3004	00:1A:A2:FA:39:90	-64	-45	-67	-61	0	-69	-59	0	-74	-54	0	-19	361	
3004	00:1A:A2:FA:39:92	-64	-45	-67	-61	0	-69	-59	0	-74	-54	0	-19	361	
3004	00:1A:A2:FA:39:95	-64	-47	-67	-61	0	-69	-59	0	-74	-54	0	-17	289	

Table 13. Localization methods: count method versus sum of squares method

O = Original, observed surveyed value, M = Measured value, L = Low (O-Range value), H = High (O+Range value), 1 = True, 0 = False, DIF = M-O, SS = $\sum DIF^2$ *. Values are in dBm.*

	Number of TRUES			SS
FID	Range 3	Range 5	Range 10	SS
3001	5	8	10	107
3002	8	10	10	77
3003	6	6	6	1013
3004	4	5	5	1628

Table 14. Outcome of counts and sum of squares

At a specific location, a certain amount of signal strength is received from the various routers (M). These are being compared with the original, surveyed signal strengths (O). In the count method, first a range of search space is being created. The larger the search space is, the more matches can be found, which leads to possible duplicate locations. In this example, if a range of ± 10 dBm would be used, one cannot tell anymore if the person finds itself at location FID=3001 or FID=3002, as can be seen in Table 14.

Count method and least SS method compared

In the least sum of squares method, large differences are contributing significantly to the elimination of unlikely locations, since they add up to the total value due to their large value. For a lower ranging value, there is no clear preference in which method can be used. However, for larger ranges, it appears that the least sum of square methods prevails, since it is capable of distinguishing locations, in the event of equal true counts. In this project, the SS method has been chosen to ensure a certain degree of certainty.

Two major weaknesses can be noted, however. A first weakness of both methods is that it requires all identifiable nodes to have an equal amount of access points, as both the count method as the sum of squares method would be disproportional to each other. In this project for example, each fingerprint must contain 10 MAC addresses. If there are only 4 MAC addresses in one fingerprint, the maximum matching value would be 4, in which this can be easily overpassed. This requires smart router placing to counter. One example includes the use of threshold and ratio counting. If, for instance, from 6 detectable routers, 4 routers are within range, then a score of 3 out of 6 (50,0%) would seem to be worse than 3 out of 5 (60,0%), despite there were more routers detectable.

A second weakness is the sensitivity of fingerprints with bad reception. If, for instance, recorded signal strengths at a certain fingerprint are bad, by default, the difference to the weakest assumed detectable signal strength (-100 dBm) is only 10 dBm. In Table 15, this is being shown: the algorithm will detect the user then at FID=3001, because of the lower SS value, although in reality, the user may find itself somewhere else.

		RSSI		SS Method		
FID	MAC	O	M	DIF	DIF²	SS
3001	00:1A:A2:C0:0E:82	-75	-100	-25	625	19250
3001	00:1A:A2:C0:0E:85	-74	-100	-26	676	
3001	00:1A:A2:FA:0F:A0	-59	-100	-41	1681	
3001	00:1A:A2:FA:0F:A2	-60	-100	-40	1600	
3001	00:1A:A2:FA:0F:A4	-60	-100	-40	1600	
3001	00:1A:A2:FA:0F:A5	-59	-100	-41	1681	
3001	00:1A:A2:FA:39:90	-41	-100	-59	3481	
3001	00:1A:A2:FA:39:92	-49	-100	-51	2601	
3001	00:1A:A2:FA:39:94	-49	-100	-51	2601	
3001	00:1A:A2:FA:39:95	-48	-100	-52	2704	
3002	00:1A:A2:FA:0F:A0	-63	-100	-37	1369	20084
3002	00:1A:A2:FA:0F:A2	-62	-100	-38	1444	
3002	00:1A:A2:FA:0F:A4	-64	-100	-36	1296	
3002	00:1A:A2:FA:0F:A5	-64	-100	-36	1296	
3002	00:1A:A2:FA:25:40	-72	-100	-28	784	
3002	00:1A:A2:FA:25:45	-72	-100	-28	78	
3002	00:1A:A2:FA:39:90	-42	-100	-58	3364	
3002	00:1A:A2:FA:39:92	-43	-100	-57	3249	
3002	00:1A:A2:FA:39:94	-43	-100	-57	3249	
3002	00:1A:A2:FA:39:95	-43	-100	-57	3249	

Table 15. Scenario for bad reception for SS method

Alternatively, if from a specific MAC no signal can be detected at all, it is possible to assign a blank value, but that would result in least sum of squares based on unequal amounts of MACs. This effectively means that unless it can be guaranteed all MACs per location are within reach.

A fifth step: sub selecting

It is therefore an additional step can then be followed, which deals with the prediction of future locations. It is implied that users do not appreciate false location information. Providing false location information when using fingerprinting is most likely to occur when unstable signal strengths are being received. For instance, a user might be prompted to be located at a floor lower or higher than he or she is, or that the device places the user at the end of the hallway, while the user has not reached that location yet. There are possible ways to cater these, by doing the following:

- Include only fingerprints of the locations along the determined route.
    - The application can be programmed in such way, only location determination can take place using the nodes (fingerprints) along the calculated route.
        - For example, for the route 2004-2003-2002-2007-2009-2011-2012, the programme only needs to check the access points coupled to these fingerprints. In the Figure 30 below, from origin to destination, the user receives a list of direction for a route. This route does not contain any signal strengths from routers B and D, so they should be eliminated.
        - However, this leads to solely filtering of routers: in the event that a location's fingerprint contains signal strengths from distant routers, the signal strengths from the excluded router will never be picked up, decreasing the probability in a correct location.

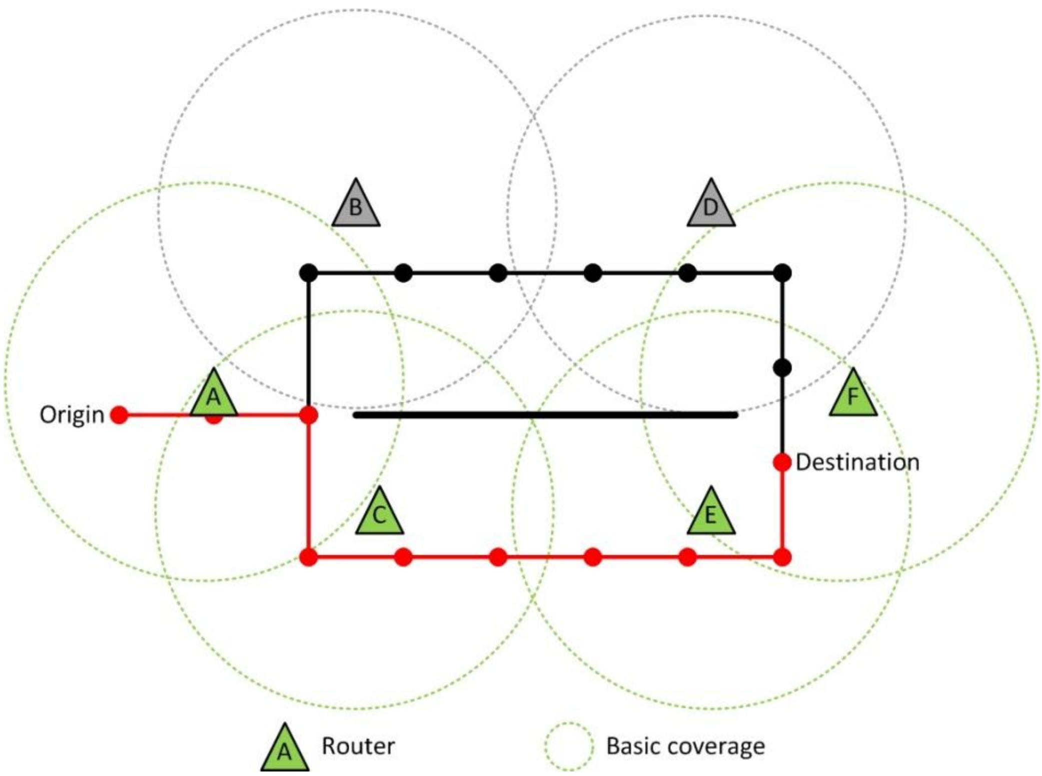


Figure 30. Exclusion of access points to narrow down search space

- Exclude fingerprints and location information that are beyond the user's current location.
    - If the fingerprints are named in such way it is recognizable which fingerprint belongs to which section or floor, it is possible to remove any access points for an area.
        - For example, upon surveying, it has been predetermined nodes are numbered in the convention <1-digit wing><1-digit floor><2-digit location>. For the route 2004-2003-2002-2007-2009-2011-2012 again, the nodes tells that these nodes are located in Wing 2, at the ground floor (0). By excluding all location information below the numbers 2000 and higher than 2100, it can be certain the user will not be promptly allocated on a wrong floor.
- If, despite the above actions have been undertaken, complete different sets are obtained than predicted as derived from the calculated route, the application has to say the user is off route, in which re-calculation of the route is proposed.

However, due to time restrictions and capability constraints, this fifth step has not been implemented in the prototype. It is therefore highly recommended that a further narrowing down of the location detection can take place. This includes one (or a combination of) the methods above described.

16.2. Geocoding and reverse geocoding

When the signal strengths and fingerprints are determined, the device can look up the ID of the fingerprint, which is coupled to both a general, navigable database (the table table_location), and a rooms database (table_rooms). The inverse occurs upon entering a room – when a user selects a room, this room is being geocoded in a fingerprint ID, which is being used to calculate a route. In this project, one room can only have one fingerprint ID, whereas one fingerprint ID can have multiple rooms. In the example of fingerprint ID (node) 3208 (see also Figure 22, page 72), rooms 2.150-2.180 all have this fingerprint. When a user would like to navigate from the Entrance/Exit to Room 2.180, the following happens:

- The user selects Entrance/Exit as departure
- The user selects Room 2.180 as destination
- The user presses Start Navigation
- The application looks up the fingerprint node for Entrance/Exit: 2001
- The application looks up the fingerprint node for Room 2.180: 3208
- The application calculates the route between nodes 2001 and 3208.
- The application provides information which nodes are required to traverse upon going from node 2001 to 3208
- The application recodes the node numbers to location information, such as "reception" and "stairs".

In the prototype, the fingerprint ID's are displayed along with the to be selected room; these can be left out in future developments.

16.3. Location determination and geocoding in Java

Box AP. 6 in the annex (A.5) contains some of the code of the location determination algorithm. First, the Wi-Fi needs to set up, which is done in lines 36-39 and lines 152-189. This allows the application to search for Wi-Fi signal strengths, along with their MACs, or Base SSIDs. Second, the to be searched

MACs have to be declared, which has been done in lines 42-43 (in Box AP. 6, only two have been displayed; one can add as many MACs as desirable). Third, the lines 47-50 creates the search space for which the received signal strength should match the recorded database, which is done in lines 209-214. The fourth step is to retrieve information from that looked up location. This can also be done via a database search, which is being carried out in lines 219-231.

In fact, the query itself exists of three parts: a first part, which includes the selection of the mac addresses and locations (line 196, also displayed below in Box 7), the middle part, in which the recorded signal strength is being compared with the lower and upper boundaries of the search space, and the closing part, which counts the matching nodes (lines 203-216), and groups it by node number (or fingerprint ID, the FID, line 197). The query is being put together, and from this new query, the FID with the maximum matches has to be the designated location (lines 229-231 to be precise).

Finally, this has to be displayed on screen, but this has to be sent in a message, rather than independent parameters, which are carried out in lines 236-256. First the parameters are being looked up and packed as a message, then the message is being decoded into visible texts again, which is carried out in lines 102-147.

```
189 public void OnRuleAchieved(RuleEvent event) {
190         WifiManager wifimanager =
191 (WifiManager)activity.getSystemService(Context.WIFI_SERVICE);
192         wifimanager.startScan();
193         List<ScanResult> ls = wifimanager.getScanResults();
194
195         //A loop to customize the query according to the measured values
196         String queryHead = "Select fid, count(*) from table_mac where ";
197         String queryTail = " group by fid";
198
199         //Vector<String> vQuery = new Vector<String>();
200         int counter = 0;
201         String queryMid = "";
202
203         if(ls.size() > 0){
204                 for(ScanResult sr: ls){
205                         //Prepare
206                         String mac = sr.BSSID;
207                         int SS = sr.level;
208                         if(counter == 0){
209                                 queryMid = "(mac  = '"+mac+"' and rssi_avg < "+(SS+4)+" and
210 rssi_avg > "+(SS-4)+")"; // First MAC with upper and lower
211                         }
212                         else{
213                                 queryMid += " or (mac  = '"+mac+"' and rssi_avg <
214 "+(SS+4)+" and rssi_avg > "+(SS-4)+")"; // Subsequent MAC with upper and lower
215                         }
216                         counter++;
217                 } //End of loop scan results
218
219                 String finalQuery = queryHead + queryMid + queryTail; // Query buildup
220
221                 int max = 0;
222                 int selectedFID = 0;
223                 Cursor cursor = db.rawQuery(finalQuery, null);
224                 if(cursor.getCount() > 0){
225                         while(cursor.moveToNext()){
226                                 int fid = cursor.getInt(0);
227                                 int count = cursor.getInt(1);
228
229                                 if(count > max){
230                                         max = count;
231                                         selectedFID = fid;
232                                 }//If some records are equal, top one is selected
```

Box 7. Snippet overview of queries for searching for nodes within the search space

16.4. Navigation algorithm

The user is able to determine a route before traveling. A simple insertion of both departure and destination can be fulfilled without the sensor technique (similar to the classic route planners before GPS was being utilized). A Dijkstra route is being implemented in the application, to preserve the simplicity of general navigation.

The Dijkstra algorithm is being used to compute the shortest path between to nodes in an edge. The algorithm searches for the shortest path from one node to all connected nodes and eliminates the paths that are longer than the currently calculated value. The algorithm will stop until the destination has been reached, in which the shortest path can be displayed.

Vogel (2009) has developed a simplistic Java code, in the sense that the Dijkstra algorithm will be used inversely for the purpose of preserving overview in the codes. The codes from Vogel (2009) are displayed in annex A.6.

In order to let the algorithm work, another Java code has to be written, in which all nodes and edges are being loaded, and from which the origin and destination nodes are being selected. This is visible in Box 8.

First, the nodes have to populate the Vertex.java. This is being done with the database recall functions, visible in lines 12-35: the FID (the fingerprint ID's: the navigable nodes) is thus being retrieved. The second step is to populate the edges for Edge.java, which is being done in lines 37-66; in this case the near entire database table is being looked up. Further processing to build the graph itself is being done in a separate class, in lines 96-105. The third step is using the information from the nodes and edges to the graph (Graph.java), which can be seen in lines 70-93: a path is being calculated using the Graph, which in turns makes use of the DijkstraAlgorithm.java (line 72-75).

Regarding intelligence in route calculation, it is possible to add extra edges between distant nodes, if they nodes follow the same route. For instance, if on a straight line between node A and B, the nodes C, D, E and F are located, this would be normally a line between A-C, C-D, D-E, E-F and F-B. It is possible to add an extra node between A and B directly, with a slightly lower impedance value assigned to it, so that upon a continuous path, directly the direction A-B can be provided. However, as the aim is to check whether the user is on route or not, this might not be the most efficient way to do so, as there can be only false checks for the intermediate points, even though in reality the user traverses those points.

```
public class RouteCalc {
        protected SQLiteDatabase db;
        protected Cursor cursor;

        private List<Vertex> nodes;
        private List<Edge> edges;

        public RouteCalc(Context c) {
                db = (new DatabaseHelper(c).getWritableDatabase());
        }

        public LinkedList<Vertex> findRoute(int fromNode, int toNode) {
                nodes = new ArrayList<Vertex>();
                edges = new ArrayList<Edge>();
                ArrayList<Integer> nodesFids = new ArrayList<Integer>();

                Cursor cursor = db.rawQuery("SELECT fid, locprop FROM table_location",
                                null);

                // Log.i("Roomlist", cursor.getCount()+"Amount found");

                if (cursor.getCount() >= 1) {
                        while (cursor.moveToNext()) {
                                Vertex location = new Vertex(cursor.getInt(0),
                                                cursor.getString(1), null);

                                nodes.add(location);
                                nodesFids.add(cursor.getInt(0));

                                Log.d("routecalc", cursor.getString(0) + " added (="
                                                + location.getRoomName() + ")");

                        }
                }
                cursor.close();

                Cursor cursor2 = db
                                .rawQuery(
                                                "SELECT cid, fid_link_01, fid_link_02,
length, direction, restricted, stairs, elevator, ramp, door, objects_left, objects_right
FROM table_connectivity",
                                                null);

                // Log.i("Roomlist", cursor.getCount()+"Amount found");

                if (cursor2.getCount() >= 1) {
                        while (cursor2.moveToNext()) {
                                addLane(cursor2.getInt(0),
                                                nodesFids.indexOf(cursor2.getInt(1)),
                                                nodesFids.indexOf(cursor2.getInt(2)),
                                                cursor2.getInt(3), cursor2.getString(4),
                                                cursor2.getString(5), false, false, false,
false,
                                                cursor2.getString(10),
cursor2.getString(11));

                                Log.d("routecalc", "Connection " + cursor2.getString(0) + "
added ... length =" + cursor2.getInt(3));
                                Log.i("routecalc", "... From "+ cursor2.getInt(1) + "=" +
nodesFids.indexOf(cursor2.getInt(1)) + " to " + cursor2.getInt(2) + "=" +
nodesFids.indexOf(cursor2.getInt(2)));
                        }
                } else {
                        Log.e("routecalc", "No connections??");
                }
                cursor2.close();

                db.close();

                Graph graph = new Graph(nodes, edges);

                DijkstraAlgorithm dijkstra = new DijkstraAlgorithm(graph);
                dijkstra.execute(nodes.get(nodesFids.indexOf(fromNode)));
                LinkedList<Vertex> path =
dijkstra.getPath(nodes.get(nodesFids.indexOf(toNode)));

                Log.e("routecalc", "Planning route from node
```

```
"+fromNode+"="+nodesFids.indexOf(fromNode));
                Log.e("routecalc", "Planning route to node
"+toNode+"="+nodesFids.indexOf(toNode));

                if (path.size() == 0) {
                        Log.e("routecalc", "No route found =(");
                }

                for (Vertex vertex : path) {
                        final String tag = "test";
                        Log.v(tag, "Route: " + vertex.getRoomName());
                        System.out.println("Route: " + vertex.getRoomName());

                }

                return path;
        }

        private void addLane(int cid, int fid_link_01, int fid_link_02, int length,
                        String direction, String restriction, boolean stairs,
                        boolean elevator, boolean ramp, boolean door, String object_left,
                        String object_right) {
                Edge lane = new Edge(cid, nodes.get(fid_link_01),
                                nodes.get(fid_link_02), length, direction, restriction,
stairs,
                                elevator, ramp, door, object_left, object_right);
                edges.add(lane);
        }
}
```

Box 8. Snippet overview of RouteCalc.java

17. Layout and results

Now the database and location determination algorithms have been established, the application itself can be constructed regarding the interface, and upon completion, tests can be performed. Chapter 17.1 deals with the design issues. Finally, in chapter 17.2, investigations of the test runs are discussed, and chapter 17.3 provides provisory remarks, which are further to be discussed in part IV.

17.1. Layout configuration

The interface consists of three screens: a screen where the user has to insert departure and destination via an AutoCompleteTextView, a screen in which the user can set route preferences (such as walker or wheelchair user, or unrestricted access), and a screen with the display of the route in descriptions. The interface is designed in XML, as shown in Box 9 for example for the Main Screen. The colour schemes are determined based on the literature recommendations as stated in chapter 12 (in particularly, sub sections 12.4 and 12.5)

As the application is a prototype, the user can only insert rooms or points of interests (POI) from the OTB building in the first insertion screen (Figure 31). If multiple building were available, another screen would be necessary to define this. The user can insert rooms by typing in a text field, which autocompletes the to be typed location. This shows the user directly what is available, and is applicable for all insert parameters (departure, destination, include via-point, exclude via-point). In the upper part of this same view, the user gets automatically to see to which room or POI he or she is located at, so he or she can use this to insert this for a departure parameter.

Figure 31. Main Screen: input and AutoCompleteTextView

In the codes below in Box 9, by default, string texts as `@string/noinfo` have been placed. As soon as a location from the gateway service has been provided, these strings will be replaced by this location information, caused by the Java code:

```
TextView textFid = (TextView) findViewById(R.id.currentType);
textFid.setText("Near: " + locprop + " (Node: " + fid + ")");
```

In this example, a node number (fid) and a location property (locprop) will be provided to replace the string with no info.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:background="#FFC000">
        <TextView
                android:layout_height="wrap_content"
                android:layout_width="match_parent"
                android:text="@string/currentlocation"
                android:textStyle="bold"
                android:textColor="#0070C0"
                android:id="@+id/CurrentLocation"></TextView>
        <TableLayout
                android:padding="3dip"
                android:layout_height="wrap_content"
                android:layout_width="match_parent"
                android:id="@+id/tableLayout1"
                android:stretchColumns="1">
                <TableRow
                        android:id="@+id/tableRow1"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content">
                        <TextView
                                android:text="="@string/noinfo "
                                android:layout_width="wrap_content"
                                android:textColor="#0070C0"
                                android:layout_height="wrap_content"
                                android:id="@+id/currentBuilding"></TextView>
                        <TextView
                                android:layout_width="wrap_content"
                                android:layout_height="wrap_content"
                                android:textColor="#0070C0"
                                android:text="="@string/noinfo "
                                android:id="@+id/currentAddress"></TextView>
                </TableRow>
                <TableRow
                        android:id="@+id/tableRow2"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content">
                </TableRow>
                <TableRow
                        android:id="@+id/tableRow3"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content">
                        <TextView
                                android:text="@string/noinfo"
                                android:layout_width="wrap_content"
                                android:textColor="#0070C0"
                                android:layout_height="wrap_content"
                                android:id="@+id/currentWing"></TextView>
                        <TextView
                                android:text="@string/noinfo"
                                android:layout_width="wrap_content"
                                android:textColor="#0070C0"
                                android:layout_height="wrap_content"
                                android:id="@+id/currentFloor"></TextView>
                </TableRow>
                <TableRow
                        android:id="@+id/tableRow3"
                        android:layout_width="match_parent"
                        android:layout_height="wrap_content">
                        <TextView
                                android:text="@string/noinfo"
                                android:layout_width="wrap_content"
                                android:textColor="#0070C0"
                                android:layout_height="wrap_content"
                                android:id="@+id/currentType"></TextView>
```

```
                    <TextView
                        android:text="@string/noinfo"
                        android:layout_width="wrap_content"
                        android:textColor="#0070C0"
                        android:layout_height="wrap_content"
                        android:id="@+id/currentRoom"></TextView>
            </TableRow>
        </TableLayout>
        <TextView
            android:id="@+id/textDeparture"
            android:layout_height="wrap_content"
            android:layout_width="match_parent"
            android:textColor="#0070C0"
            android:text="@string/departure"
            android:textStyle="bold"></TextView>
        <AutoCompleteTextView
            style="@style/FakeTextField"
            android:id="@+id/inputDeparture"
            android:background="@drawable/textlines"
            android:clickable="true"
            android:textColor="#0070C0"></AutoCompleteTextView>

        <TextView
            android:id="@+id/textDestination"
            android:layout_height="wrap_content"
            android:layout_width="match_parent"
            android:textColor="#0070C0"
            android:text="@string/destination"
            android:textStyle="bold"></TextView>
        <AutoCompleteTextView
            style="@style/FakeTextField"
            android:id="@+id/inputDestination"
            android:background="@drawable/textlines"
            android:clickable="true"
            android:textColor="#0070C0"></AutoCompleteTextView>


        <TextView
            android:id="@+id/textExample"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textColor="#0070C0"
            android:text="@string/specificationExample"
            android:textSize="10sp"
            android:paddingBottom="10sp"></TextView>
        <Button
            android:layout_height="wrap_content"
            android:layout_width="match_parent"
            android:id="@+id/buttonNavigationStart"
            android:text="@string/startnavigation"></Button>
        <Button
            android:layout_height="wrap_content"
            android:layout_width="match_parent"
            android:id="@+id/buttonSettings" android:text="@string/settingsLabel"></Button>
</LinearLayout>
```

Box 9. XML file of the Main Screen

In the second screen, the user is able to define the route parameters. This is being done with radio buttons for the type of user, and an enabler button for the restriction level. Enhanced features can be placed here as well, such as a text-to-speech engine, or the display of the graph structures.

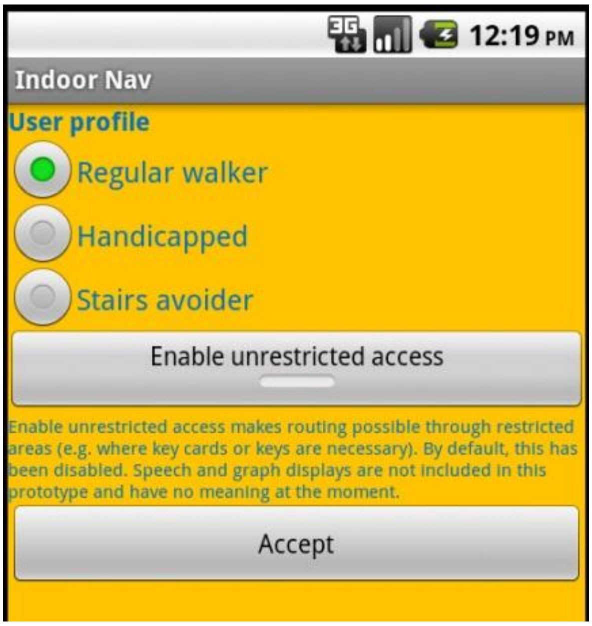


Figure 32. Settings screen

In the third and final screen, the user should see the route, along with the current location. As the application involves a non-geometry feature, no maps are to be displayed. However, the user gets to see if he or she is at the correct place according to the provided route. In the prototype, the node number is being displayed as reference. Because the Android Emulator does not support Wi-Fi measuring, Figure 33 below shows no information about the current position.

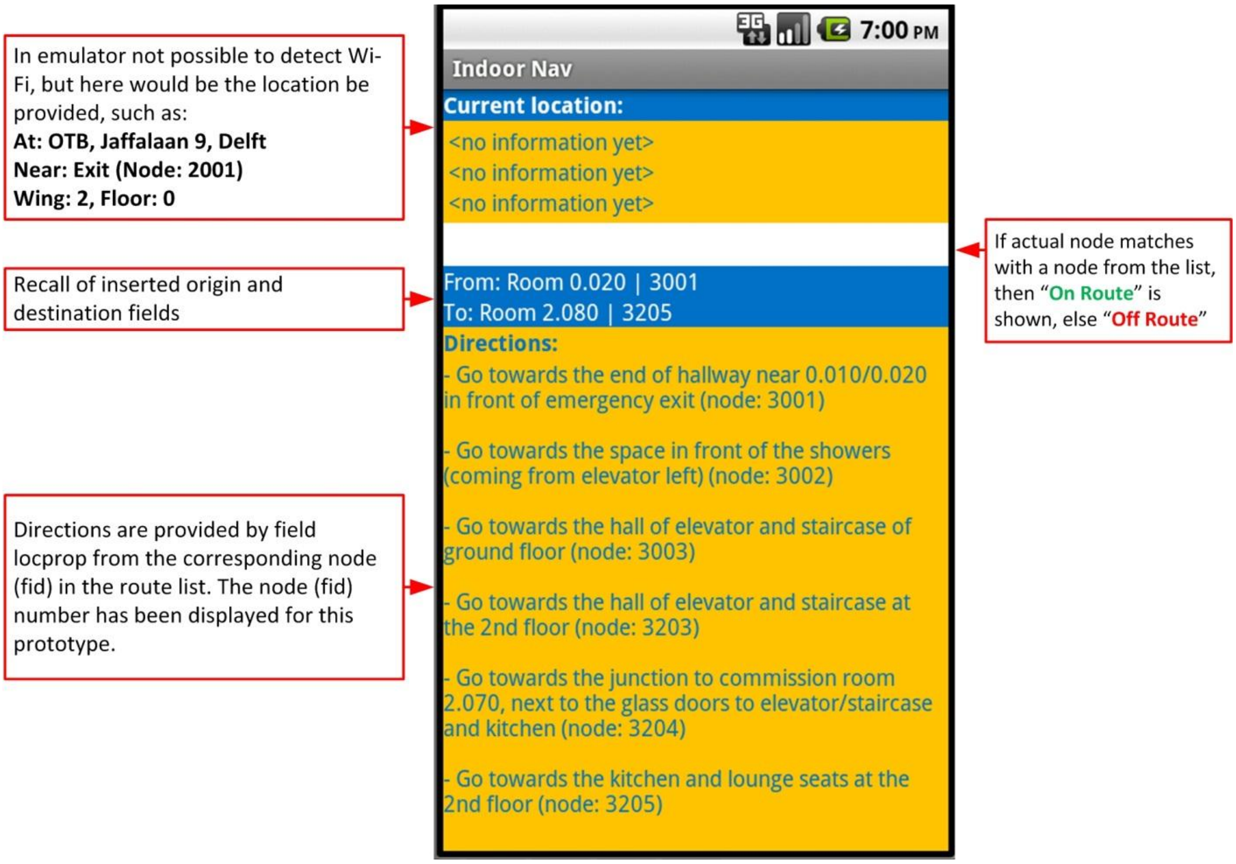


Figure 33. Navigation screen: output of a route

17.2. Testing the application

Upon completion of the application, test runs have been performed, of which the methodology and results are discussed in this sub chapter. The testing includes application validation (17.2.1), the location check (17.2.2) and the navigation itself (17.2.3).

17.2.1. Application validation

The functioning of the application itself is the first part of testing, where Wi-Fi does not necessarily have to be used. This includes the input, processing and output mechanisms without Wi-Fi, as well as the layout and the interface. For instance, a user might plan ahead and use the application to plan a journey from the entrance to room 2.250. The application validation is the major part of the development, and has to work in advance, before a location check comes in.

The prototype has been configured in such way that the node numbers are still visible. That means that the user can search for the FID number of the nodes/locations. A built-in feature is that it will also check for empty or incorrect fields: it is not possible to calculate a route between points that do not exist. Further extensions have been taken into account, such as the possibility to calculate routes based on user preferences, but are not operational as it is not the aim of the research to look at various outcomes of user profiles, but rather on localization of the user.

17.2.2. Criteria of location check and test performances

In the prototype, the user first sees what the current location is. This is displayed in the top of the screen. At various locations, this has been tracked at twenty different timestamps. The location is considered correct, if the application states that the user is exactly at the right place, or at the place adjacent to it (with a maximum of two nodes), since the signal strengths may vary over time and space. This does not apply for a wrong placement on the vertical dimension, since the signal strengths are significantly different from each other at different heights.

The filtering of search space by ± 4 dBm has been maintained in the next scan results. The first scan compromised the registration of 12 location nodes, in 20 consecutive time stamps. The results have been distinguished by:

- Close matches:
  - Location has an exact match with the database
  - Location was within one node away, at the same floor
    - E.g. if the actual location is node 3109, then 3108 and 3110 are valid.
  - Location was within two nodes away, at the same floor
    - E.g. if the actual location is node 3109, then 3107, 3108, 3110 and 3111 are valid.
- Bad matches:
  - Location was within two nodes away, but at a different floor
    - E.g. if actual location is 3109, then 3008-3011, 3208-3211 are valid
  - Location was more than two nodes away, but at the same floor
  - Location was more than two nodes away, and at a different floor
  - Location was physically within two nodes away, but at different part of the building
    - This occurs in places with many complex building parts, in which the user is prompted to be near a location close to another wing, for example.

Two major scans have been performed. In the first series, location information was being retried based on fingerprints composed by the ten strongest signal strengths at that point, as originally proposed. This also includes routers that are on a different floor than where the users are. The second series of scan results were based on eight MAC addresses, whose access points located on the same floor. In this extent, the physical space of the router is being combined with the physical location of the fingerprint. This has been made possible, since the routers at the OTB contain four MAC addresses. To illustrate the routers' locations and their reaching space on that same floor, see Figure 34:

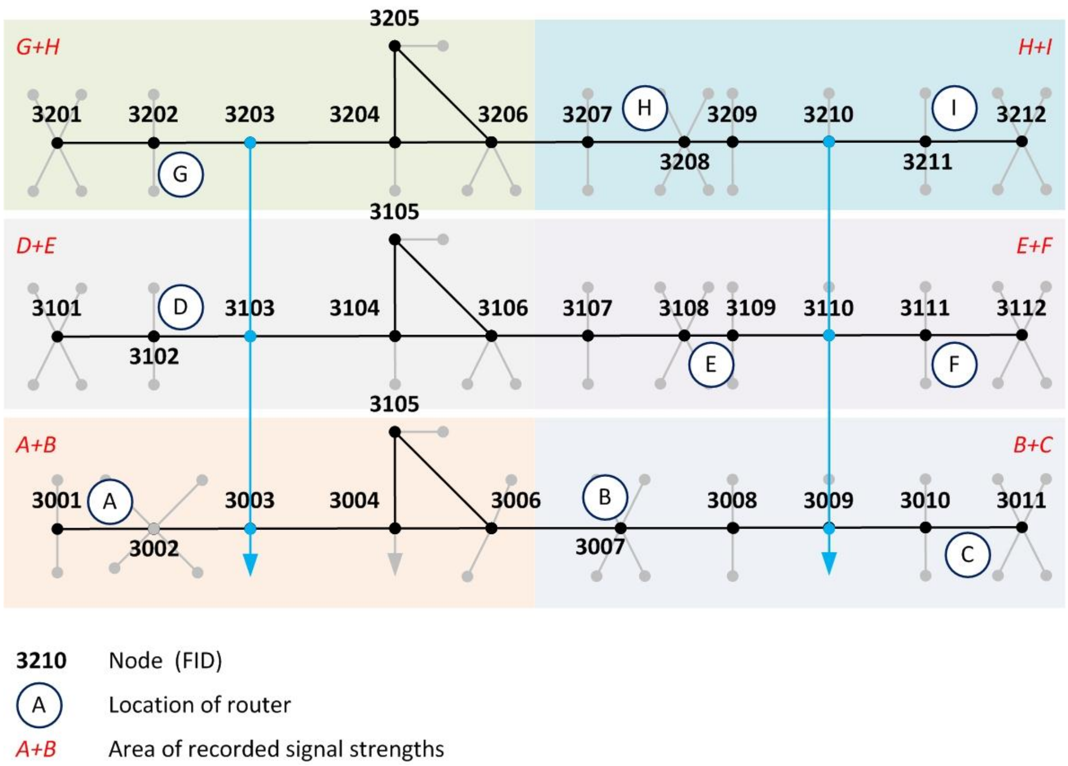


Figure 34. New database method: using coverage areas instead of storing best signals

Next, at 12 distinctive locations, scans were performed. The following locations were chosen (Table 16):

Node	Location	Remark
2002	Wing 2, Ground Floor, near main exit	Open area near nodes 2006-2007-2008
2005	Wing 2, Ground Floor, near back exit	Small corridor near end
2008	Wing 2, Ground Floor, corner of two rooms	Open area near nodes 2002-2006-2007
3002	Wing 3, Ground Floor, near emergency exit	End of hallway, near router A
3007	Wing 3, Ground Floor, centre point	Near router B, centre of corridor
3011	Wing 3, Ground Floor, end of hallway	Near router C
3101	Wing 3, Floor 1, end of hallway	Near router D
3105	Wing 3, Floor 1, near kitchen	Far away from router D and E
3110	Wing 3, Floor 1, second staircase	Between routers E and F
3203	Wing 3, Floor 2, primary staircase	Between routers G and H
3209	Wing 3, Floor 2, next to glass doors of corr.	Near router H
3211	Wing 3, Floor 2, near end of hallway	Near router I

Table 16. Overview of test nodes for location checks

Results on overall scan

Out of the 240 obtained values, only 14 of them had exact matches, which is a mere 5.8% of the scan results. Moreover, 66 were within two nodes of the actual location, which compromises a 27.5% of the scan results. This compromises that the user was in one third (33.33%) at or near the appropriate location. In 47 cases (19.6%), the user was located within two nodes of the actual location, although it was on a different floor. This was mostly being the case when the user was at the ground or first floor. This suggests that signal strengths were matched from the floor or below the user, as this rarely happened on the second floor, where only one floor below could be matched. Still, 26.3% was matched on the same floor, but beyond two nodes of where the user actually was.

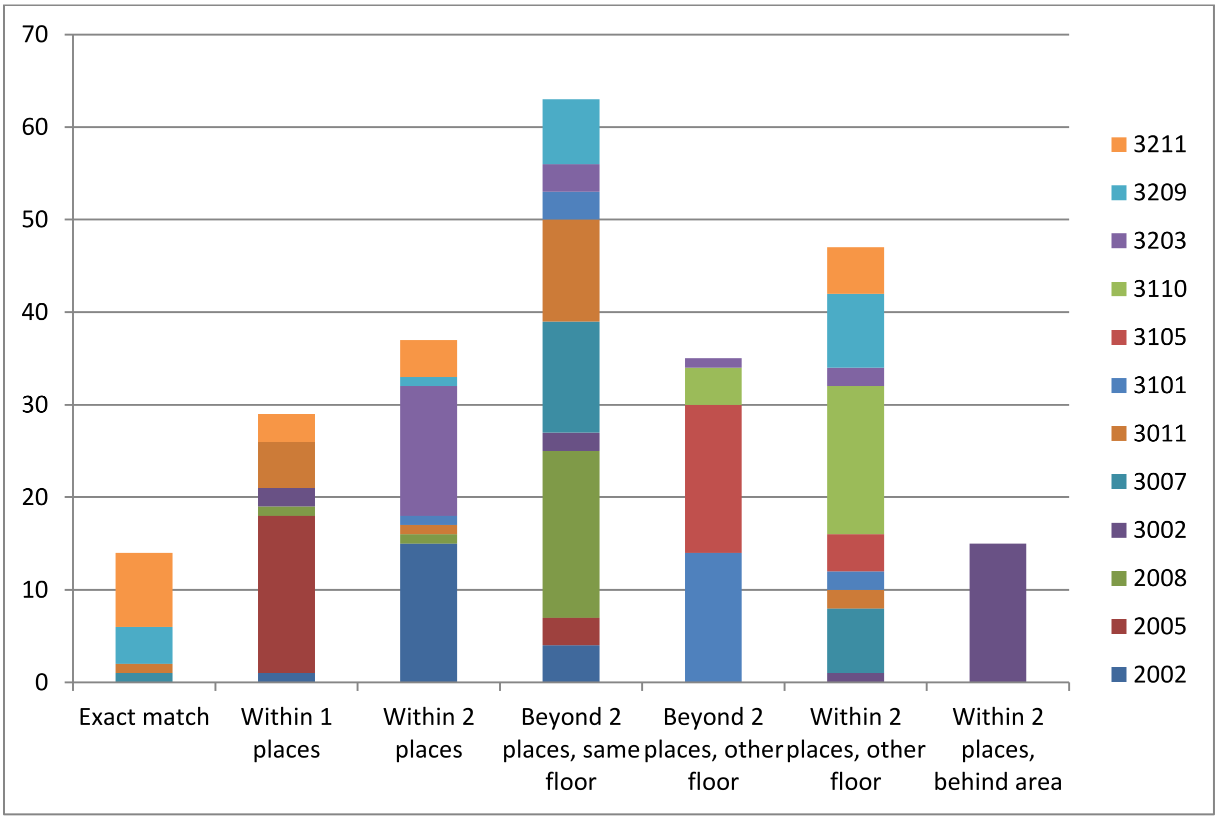


Figure 35. First scan results - grouped by match, specified by location (fingerprint) node, 12 nodes, wing 2+3.

If a closer look is being taken on the results of solely Wing 3 in the test case, it seems that the wrong allocation on the wrong floor has been affected by the scan results of the locations in Wing 2. Simultaneously, the total cases of a near correct match dropped to an exact 25% (45 out of 180; with Wing 2 included, which compromised 80 out of 240). This is caused by the near correct placements of node 2002 and 2005: 15 out of 20 time intervals, the user was located within 2 places away of node 2002, whereas it happened 17 out of 20 time intervals the user was located within 1 place away of node 2005.

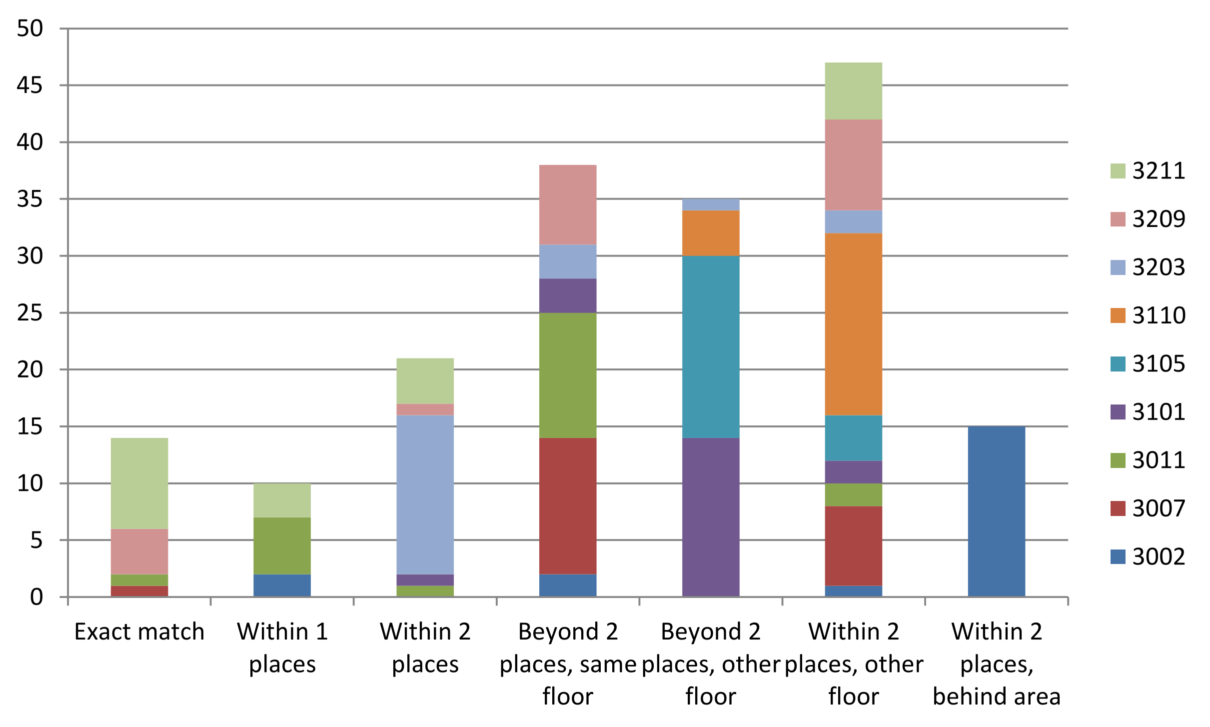


Figure 36. First scan results - grouped by match, specified by location - wing 3 nodes

The outcomes are also displayed in Table 17 below, and further specified for Wing 3 only in Table 18.

Node	3002	3007	3011	3101	3105	3110	3203	3209	3211	2002	2005	2008	TOTAL
Close matches:	**2**	**1**	**7**	**1**	**0**	**0**	**14**	**5**	**15**	**16**	**17**	**2**	**80**
% From total observations	**0,8%**	**0,4%**	**2,9%**	**0,4%**	**0,0%**	**0,0%**	**5,8%**	**2,1%**	**6,3%**	**6,7%**	**7,1%**	**0,8%**	**33,3%**
Exact match	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**4**	**8**	**0**	**0**	**0**	**14**
% From total observations	**0,0%**	**0,4%**	**0,4%**	**0,0%**	**0,0%**	**0,0%**	**0,0%**	**1,7%**	**3,3%**	**0,0%**	**0,0%**	**0,0%**	**5,8%**
Within 1 places	2	0	5	0	0	0	0	0	3	1	17	1	**29**
Within 2 places	0	0	1	1	0	0	14	1	4	15	0	1	**37**
% From total observations	0,8%	0,0%	2,5%	0,4%	0,0%	0,0%	5,8%	0,4%	2,9%	6,7%	7,1%	0,8%	**27,5%**
Out of match:	**18**	**19**	**13**	**19**	**20**	**20**	**6**	**15**	**5**	**4**	**3**	**18**	**160**
% From total observations	**7,5%**	**7,9%**	**5,4%**	**7,9%**	**8,3%**	**8,3%**	**2,5%**	**6,3%**	**2,1%**	**1,7%**	**1,3%**	**7,5%**	**66,7%**
Beyond 2 places, same floor	2	12	11	3	0	0	3	7	0	4	3	18	**63**
% From total observations	0,8%	5,0%	4,6%	1,3%	0,0%	0,0%	1,3%	2,9%	0,0%	1,7%	1,3%	7,5%	**26,3%**
Beyond 2 places, other floor	0	0	0	14	16	4	1	0	0	0	0	0	**35**
Within 2 places, other floor	1	7	2	2	4	16	2	8	5	0	0	0	**47**
Within 2 places, behind area	15	0	0	0	0	0	0	0	0	0	0	0	**15**
% From total observations	6,7%	2,9%	0,8%	6,7%	8,3%	8,3%	1,3%	3,3%	2,1%	0,0%	0,0%	0,0%	**40,4%**

Table 17. Scan results attempt 1

Node	3002	3007	3011	3101	3105	3110	3203	3209	3211	TOTAL
Close matches:	**2**	**1**	**7**	**1**	**0**	**0**	**14**	**5**	**15**	**45**
% From total observations	**1,1%**	**0,6%**	**3,9%**	**0,6%**	**0,0%**	**0,0%**	**7,8%**	**2,8%**	**8,3%**	**25,0%**
Exact match	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**4**	**8**	**14**
% From total observations	**0,0%**	**0,6%**	**0,6%**	**0,0%**	**0,0%**	**0,0%**	**0,0%**	**2,2%**	**4,4%**	**7,8%**
Within 1 places	2	0	5	0	0	0	0	0	3	**10**
Within 2 places	0	0	1	1	0	0	14	1	4	**21**
% From total observations	1,1%	0,0%	3,3%	0,6%	0,0%	0,0%	7,8%	0,6%	3,9%	**17,2%**
Out of match:	**18**	**19**	**13**	**19**	**20**	**20**	**6**	**15**	**5**	**135**
% From total observations	**10,0%**	**10,6%**	**7,2%**	**10,6%**	**11,1%**	**11,1%**	**3,3%**	**8,3%**	**2,8%**	**75,0%**
Beyond 2 places, same floor	2	12	11	3	0	0	3	7	0	**38**
% From total observations	1,1%	6,7%	6,1%	1,7%	0,0%	0,0%	1,7%	3,9%	0,0%	**21,1%**
Beyond 2 places, other floor	0	0	0	14	16	4	1	0	0	**35**
Within 2 places, other floor	1	7	2	2	4	16	2	8	5	**47**
Within 2 places, behind area	15	0	0	0	0	0	0	0	0	**15**
% From total observations	8,9%	3,9%	1,1%	8,9%	11,1%	11,1%	1,7%	4,4%	2,8%	**53,9%**

Table 18. Scan results attempt 1 - Wing 3 only

Second test run

In the second test run, the same locations in were being compared, but this time the focus was only on wing 3. In addition, the database was reconfigured and remeasured (new database) with the same device as that it is being read, to marginalize the errors in the database caused by using a high-end receiver. Instead of an average scan for the best 10 MAC addresses, only the routers on the same floor as the location node has been used (see Figure 34), to limit the search space. The rationale is that it will reduce the probability of placing the user on the wrong floor, as the signals that come through ceilings and floors have been much weakened as such. From the 9 x 20 = 180 scan results, 81 times the user was located near the appropriate location, from which 46 times was at the exact location as predicted (=25.6% of the total 180, or 56.8% from all 81 good matches, see also Table 19). This is an increase of 20% for all (near-)correct nodes. The exact matches have been increased from 7.8% to 25.6%.

The results at locations which are situated near the end of the hallway or at open spaces are prone to false location conversions, when taking a closer look at the direct results (see Annex C.1). This is most likely being the cause of the availability of only the strong signals from a nearby router, whereas the weaker signals of faraway routers are hard to obtain. In open spaces, it is possible that many other signals from floors below are easier to receive, as the signals elsewhere need to penetrate ceilings and floors first, which results in much weaker signals to be interfering the scanning results. For example, it has been found that there were no fingerprints recorded at the intermediate floor levels in the staircases, making comparisons not possible. Being halfway, the device picks up signal strengths and compares it with the database, and places the user on a floor (while the user finds itself between floors), or in a wrong wing if the location is not too far away from a different wing.

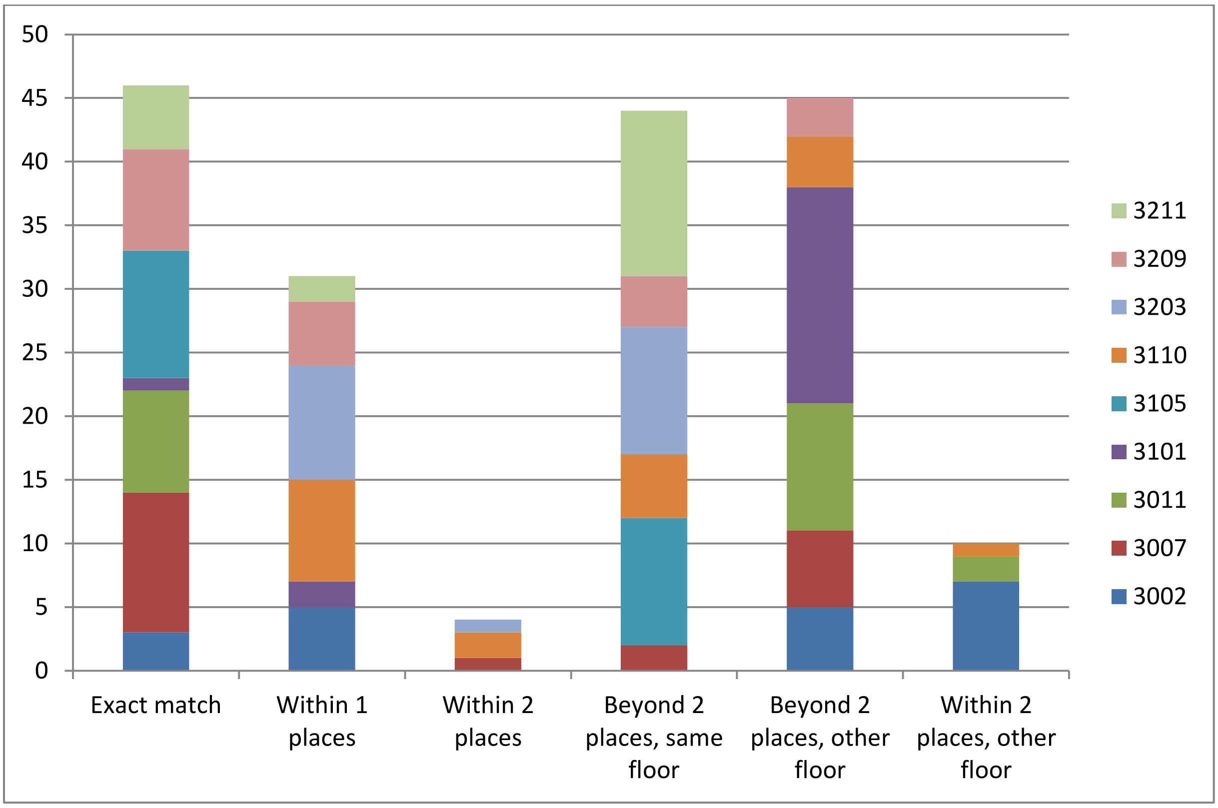


Figure 37. Second scan results – grouped by match, specified by location - wing 3 nodes

Node	3002	3007	3011	3101	3105	3110	3203	3209	3211	TOTAL
Close matches:	**8**	**12**	**8**	**3**	**10**	**10**	**10**	**13**	**7**	**81**
% From total observations	**4,4%**	**6,7%**	**4,4%**	**1,7%**	**5,6%**	**5,6%**	**5,6%**	**7,2%**	**3,9%**	**45,0%**
Exact match	**3**	**11**	**8**	**1**	**10**	**0**	**0**	**8**	**5**	**46**
% From total observations	**1,7%**	**6,1%**	**4,4%**	**0,6%**	**5,6%**	**0,0%**	**0,0%**	**4,4%**	**2,8%**	**25,6%**
Within 1 places	5	0	0	2	0	8	9	5	2	**31**
Within 2 places	0	1	0	0	0	2	1	0	0	**4**
% From total observations	2,8%	0,6%	0,0%	1,1%	0,0%	5,6%	5,6%	2,8%	1,1%	**19,4%**
Out of match:	**12**	**8**	**12**	**17**	**10**	**10**	**10**	**7**	**13**	**99**
% From total observations	**6,7%**	**4,4%**	**6,7%**	**9,4%**	**5,6%**	**5,6%**	**5,6%**	**3,9%**	**7,2%**	**55,0%**
Beyond 2 places, same floor	0	2	0	0	10	5	10	4	13	**44**
Beyond 2 places, other floor	5	6	10	17	0	4	0	3	0	**45**
Within 2 places, other floor	7	0	2	0	0	1	0	0	0	**10**
% From total observations	6,7%	4,4%	6,7%	9,4%	5,6%	5,6%	5,6%	3,9%	7,2%	**55,0%**

Table 19. Scan results attempt 2 - Wing 3 only

17.2.3. Indoor Wi-Fi Navigation testing

The navigation, with the Wi-Fi fingerprints as location check, is the core of the application. The user has planned a journey, and the position is being monitored with the Wi-Fi fingerprints. The user has to follow the rules and directives provided and should receive a message whether the user is on or off route.

Routes are being calculated with the graph and Dijkstra algorithm. Neutral directions are provided, along with the node numbers. If the device picks up the signals and translates that into a node ID, this is being compared with the list of nodes in the calculated route. If this ID is within the list, the message “on route” is being displayed, if not, “off route” is being displayed.

Within wing 3 of the OTB building, various routes have been traversed using the application. Five different type of routes have been travelled upon in different travel directions, as displayed in Table 20. For each route, the number of “off route” messages have been registered, as well as all node numbers that the device picks up during the journey, which is being compared with the provided list.

Upon registration of the correctness of the information provided, the following scenarios are being used:

- The application detects the user at a location that is provided in the list, and the user finds himself at the location that should match the list;
  - Example: user is at location FID=3001; the device provides message user is at FID=3001.
  - This will be registered as fully correct (in Table 20 denoted as “on” – on route).
- The application detects the user at a location that is provided in the list, but the user is not at the location that he should be at;
  - Example: user is at location FID=3001; the device provides message user is at FID=3002.
  - This will be registered as partly correct (in Table 20 denoted as “on/off”).
- The application detects the user at a location that is not provided in the list, and the user does not find himself at that location;
  - Example: user is at location FID=3001; the device provides message user is at FID=2004.
  - This will be registered as incorrect (in Table 20 denoted as “off” – off route).

As expected, the results in correct route confirmation are similar to those as the general location confirmation, since the checks are based on location, although in this case, the test run involved a swift FID acquisition (maximum of 1 second waiting at a location), instead of waiting for a long acquisition time (20 seconds as in the previous section). In 41.3% of all route confirmation provisions, a correct message has been given. Upon examining the variations between the different routes, it seems that the results were poor on the route that was fully from one end to another on the first floor (route 2), as well as the route that dealt with travelling within the north end (route 5). From the former, this can be explained that despite using routers within the same physical space, the device still picks up signal strengths from floors above or below the user. Since the probability a signal strength from a floor below the bottom most or above the top most floor is nil, that comes in no surprise.

Series 1											
Route	Characteristic	From	Via	To	Nodes	On	On/off	Off	On	On/off	Off
1	South End 0F - North End 2F	3005	3009	3212	9	5	2	2	**55.6%**	22.2%	22.2%
2	South End 1F - North End 1F	3101	3107	3112	11	4	3	4	**36.4%**	27.3%	36.4%
3	South End 2F - North End 1F	3201	3110	3112	12	6	4	2	**50.0%**	33.3%	16.7%
4	South End 0F - South End 2F	3001	3103	3201	7	2	1	4	**28.6%**	14.3%	57.1%
5	North End 2F - North End 0F	3212	3110	3011	7	2	3	2	**28.6%**	42.9%	28.6%
Series 2											
Route	Characteristic	From	Via	To	Nodes	On	On/off	Off	On	On/off	Off
1	South End 0F - North End 2F	3005	3009	3212	9	4	3	2	**44.4%**	33.3%	22.2%
2	South End 1F - North End 1F	3101	3107	3112	11	4	2	5	**36.4%**	18.2%	45.5%
3	South End 2F - North End 1F	3201	3110	3112	12	5	5	2	**41.7%**	41.7%	16.7%
4	South End 0F - South End 2F	3001	3103	3201	7	3	0	4	**42.9%**	0.0%	57.1%
5	North End 2F - North End 0F	3212	3110	3011	7	4	0	3	**57.1%**	0.0%	42.9%
Series 3											
Route	Characteristic	From	Via	To	Nodes	On	On/off	Off	On	On/off	Off
1	South End 0F - North End 2F	3005	3009	3212	9	3	3	3	**33.3%**	33.3%	33.3%
2	South End 1F - North End 1F	3101	3107	3112	11	4	3	4	**36.4%**	27.3%	36.4%
3	South End 2F - North End 1F	3201	3110	3112	12	5	1	6	**41.7%**	8.3%	50.0%
4	South End 0F - South End 2F	3001	3103	3201	7	4	0	3	**57.1%**	0.0%	42.9%
5	North End 2F - North End 0F	3212	3110	3011	7	2	0	5	**28.6%**	0.0%	71.4%
Total											
Route	Characteristic	From	Via	To	Nodes	On	On/off	Off	On	On/off	Off
1	South End 0F - North End 2F	3005	3009	3212	27	12	8	7	**44.4%**	29.6%	25.9%
2	South End 1F - North End 1F	3101	3107	3112	33	12	8	13	**36.4%**	24.2%	39.4%
3	South End 2F - North End 1F	3201	3110	3112	36	16	10	10	**44.4%**	27.8%	27.8%
4	South End 0F - South End 2F	3001	3103	3201	21	9	1	11	**42.9%**	4.8%	52.4%
5	North End 2F - North End 0F	3212	3110	3011	21	8	3	10	**38.1%**	14.3%	47.6%
Total					**138**	**57**	**30**	**51**	**41.3%**	**21.7%**	**37.0%**

Table 20. Results of route confirmation results within wing 3

A different approach upon testing the route confirmation is not by checking the correct location confirmation, but on the way the route guidance is provided along the way. In this project, a simple topological route network guidance has been provided. Without the presence of geometry, it is hard to tell whether the user has to turn left or right, and therefore it is hard to tell whether the user's orientation is correct. This can be relieved if information about correctness is involved. For instance, if the user traverses a floor from one end to another, he can get confirmation such as "You are doing correct if you see the add office room numbers are adding up". There is still no geometry involved, but still some sort of route confirmation is present. Further research is desirable to do so.

After a short demonstration, test persons have noted many expansions are worthy, which includes the addition of photos, graphs, images, maps and pictograms. In addition, it was noted that the text output was rather long or confusing. This confirms the weakness of providing neutral text based directions. As there is no geometry involved, it is rather difficult to tell whether the user has to turn left or right. The next step of navigation would indeed be implementing geometric features.

17.3. Remarks and conclusions on the application

How can Wi-Fi fingerprinting determine a location which does not utilize geometric features, and how can it be implemented in an application as such?

Not utilizing geometric features requires the developer to realize that a navigable graph network must emphasize on the topology and structure that exist between navigable places.

First of all, the results are influenced by the type of receiver that was being used to record the data itself. A high-end receiver such as a laptop provides better scan results than the middle-end smartphone receiver, which was visible from the fingerprint database. As a result, the device that is being used for the application might over- or underestimate the recorded scan results. It means that the fingerprint database needs frequent updates, and that there are multiple databases necessary, for each class of end receivers.

Second, the location methodology is a simplified version of map-matching algorithms, in the sense it does not uses geometric features. A least sum of square is hard to use, since Android automatically assigns either non-existent values (-100 dBm) or blank values, which are in both cases sensitive to providing false location. The count method location determination itself might not be fair as well, since it only checks the range from the signals that have been recorded and evaluated. As such, stronger signals have an equal weight as weak signals. With that regard, smart localization should be implemented.

Third, results on the localization is influenced by the database records: better results have been found when only MACs from the same physical space have been used in the recording phase, although that is not a guarantee: still less than 50% provides a correct, or near-correct estimation.

A few suggestions about the application and its development are mentioned: First, automatizing the process of surveying can be done with the smartphone itself. In the first place, the signal strengths can be recorded, but by building an application in which the surveyor can add location information to this signal strength, a graph (and thus navigable network) can be created.

Second, a combination of other techniques could improve the localization method, as well as the navigation module. For instance, an accelerometer, which is also present at present-day smartphones, can help detect orientation aspect, and from there, clearer route descriptions can be given, instead of neutral descriptions. As Wi-Fi is rather unstable due to the variations in signal strength (see annex C.1 and C.2), further research should be carried out to solve this.

PART IV - Conclusions

Contents

In this fourth and final part, the output is being discussed. In chapter 18, the final conclusions are being drawn and the main research question *is it possible to develop an indoor navigation service for a mobile platform, with the use of Wi-Fi fingerprinting technology and Open Location Services standards, but without using geometry?* will be answered. The results are further being evaluated with its limitations and recommendations in the chapters 19 and 20 respectively.

18. Final Conclusion

In this thesis project, attempts have been made to investigate the possibilities of indoor navigation on a smartphone, with the use of Wi-Fi technology, but without using building geometry. The research question was:

Is it possible to develop an indoor navigation service for a mobile platform, with the use of only Wi-Fi fingerprinting technology and the framework of the Open Location Services standards, but without using building geometry?

It is possible to launch an indoor navigation service for a mobile platform, with Wi-Fi fingerprinting technology. The framework of the Open Location Services were provided as a guideline to set up the application, but they remained unused, as they cannot be combined with a project which does not use geometric features such as maps, since the parameters are focussed on that aspect. However, the location predictions are too unreliable to be used.

Caution has to be taken upon fingerprinting, since fingerprinting assumes that the characteristic signal strengths are (1) dependent on physical objects present in the navigable space due to multipath and (2) are changing over time. In addition, their transmitters, the routers or Media Access Control units (MAC), can be replaced as well. Upon site surveying, it is highly recommended to store at least a variety of MACs, along with their respective signal strength, at specific locations. In addition, great care has to be taken upon registering the MACs themselves, since they have to be combined with location information, which indirectly leads to identifiable persons. This, in turn, can be prohibited by law in countries such as the Netherlands. Indoor navigation should thus be limited to public buildings only.

Two types of location determination have been proposed: a count method, and the least sum of squares method. In Android, both methods are constrained by how access points out of reach are being treated: non-detectable access points are either assigned with a value of -100 dBm, or as a blank value. Regarding the least sum of squares (SS) method, in the former case this will lead to the promotion of recorded weak signal strengths at certain locations. In the latter case, this will lead to unequal SS value to compare, as one location might have a SS value based on five access points, and another location might have a SS value based on seven access points, in which it is most likely the location with the fewest access points prevails. The count method will solely check for matches in a defined search space. This includes a search margin of roughly 3-5 dBm. In this project the value of 4 dBm as a search space margin has been used.

It has been seen that upon using recorded values from the ten best access points within reach, often false locations have been provided. In a short live scanning survey, only 25.0% of all 180 measurements were within an acceptable predicted location, which equals the location being perceived within a maximum of two nodes away, at the same floor. In only 7.8% of all 180 cases, the location was exactly the same. In 53.9% of all cases, the location estimation was on the wrong floor and far away. When the database is being replaced with only MAC addresses from access points in the same physical space (= same floor) as the fingerprint/node, the results seems to improve: 45.0% of all 180 measurements was now within the acceptable predicted location, and 25.6% of all measurements were perceived at the exact same spot as where the device was. This has been at the expense of the estimations far away on wrong floors: this percentage has been decreased to 30.5% of all cases.

Further conclusions can be drawn regarding the localization algorithm and the database. First of all, it is desirable that the results obtained from the site surveying should be done with the same device as the live scanning results, which has been done in the second test runs. This is to prevent the database to be filled with recordings that are too high or too low, in which the amount of over- and underestimation can be diminished. Second, certain locations are highly likely to provide false location information, as they are dependent on the building structure, and thus highly influent by multipath effects. These locations where fingerprints are stored, includes locations at staircases, or near the end of the hallway. However, it is rather difficult to realize stable location detection.

The navigation algorithm being used, the Dijkstra algorithm, can be further expanded, with options regarding user profiles. The navigation output itself has to be improved, since it now uses neutral descriptions, as there is no orientation aspect, except for recognizable objects that are being used to provide directions. This is clearly one of the weaker aspects of the project, since it does not use building geometry at all. It is recommended that previous or future locations, should also be used to provide route guidance to the user. Furthermore, paths that are in the same direction of extension are now all treated as single paths. Smart routing in the sense of providing directions such as ‘go to the end of the hallway near point E’ instead of ‘go towards point A, B, C, D, E’ is highly recommended. With that respect, the integration of additional technologies, such as a accelerometer could provide a useful extension, as it will help to incorporate geometry after all.

The representation is also difficult to improve when one considers solely text-based navigation. Enhances in the application interface are desirable, such as enabling text-to-speech, routing by user profiles, displaying graphs and active maps, and estimated times of arrival, includes better route handling and narrowing down location determination algorithms and: this prototype checks whether the determined location matches with a calculated route, and states whether the user is on or off route. Ideally, the route has to be recalculated if the user does not find itself at its path predicted. In addition, by narrowing the location determination, such as ‘only looking in the search space at the same floor’ or ‘look at only three nodes away from me’ further upgrades the prototype. The incorporation of other localization techniques, along with its enhancements (such as an accelerometer), would be helpful.

To answer the main research question, a simple no would suffice, as summed up:

- Wi-Fi signals are instable and fluctuate significantly;
- Quality of fingerprinting depends on quality of receiver for recording versus live reading;
- Fingerprinting databases requires frequent updating;
- Emphasis on topology is possible without geometric features, but orientation aspects are difficult to solve;
- Parameters of the OpenLS cannot be used, since they are dependent on geometric features, although their framework can be used;
- Localization methodologies dependent on how Android treats out of reach routers, and on the how the amount of mac addresses being read are treated equally;
- It is proven that despite changing the received signal strengths in favour of the device, below 50% of all location estimations were correct, making the reliability of the whole application poor.

19. Reflection: discussion, limitations and recommendations

In this chapter, limitations and recommendations are being discussed. First, limitations are addressed in section 19.1, followed by recommendations in section 19.2 regarding the application itself, and in section 19.3 regarding localization methods.

19.1. Limitations in general

Logging (as in recording the users complete path, not to be confused with tracking, which is following the users path) and the maintenance are not included in this research, since it is not the goal to allow users to track their progress, but rather navigating them from one end to another. However, tracking could be useful in combination with pedometer-like application for instance. Finally, since it is the purpose to create a prototype, no maintenance is required either.

Unlike the outdoor navigation, data on indoor locations is not widely available. There are no vendors for networks such as FalkPlan and TeleAtlas, and not all companies are willing to provide floor plans for public use, or even if there is a provision, it might be possible that not everything is covered to keep it secret. This might cause less accuracy, or missing data. On the other hand, since the non-geometry is the keyword to this thesis, it might be possible not to work with maps at all, therefore limiting the data usage in an advantageous way.

Fingerprinting requires a lot of pre-processed work, in the form of site surveying. Also, the fact that it is capable of handling multi-path makes it a burden at the same time. As soon as large objects, including people, are moved, the fingerprints will surely be affected. The same applies for the fixation of MAC addresses. Since the received signal strengths are dependent on the MAC address, patterns might not be found, as soon as an access point has been replaced by another. An application thus has to be constantly updated, and from time to time a new survey has to take place. Warnings about imprecise location estimation as such should be present.

Text descriptions itself are useful, as they are capable of steering the user on what he or she has to do. However, as Hsu (2011) already pointed out, is that the attention of users is in the visualization. Photos, pictograms and maps are in that sense better options. The application can thus further enhanced with visual media as such. Yet, when maps are being used, the concept of localization with fingerprinting must be changed to the concept of positioning with map matching, which requires different implementation algorithms. Ideally, the reverse geocoding from fingerprints to location information can be re-geocoded again, to known, absolute locations. In that sense, indoor navigation can be further enhanced, and it is highly recommended to do so in future research.

The aim of this thesis was to investigate possibilities regarding indoor Wi-Fi without building geometry. It was not the intention to develop a full and complete product, but rather on how to use the various aspects of navigation. As such, the result is that the location determination, and the navigation are standalone versions as well. Ideally, those two have to be combined for real live navigation as one is known for car navigation, with the addition of the confirmation of where the user is – either on route, or off route. If this incentive can be reached, the use of maps or graphs can be further enhanced, as the device might show the user where the location on the graph is.

19.2. Regarding the application

More recommendations can be discussed for the prototype itself.

A first recommendation is that the database should be stored on a remote server, so that users will not need to update the application constantly regarding changes of macs and signals. The downside however, is that it requires at least a 3G wireless connection, in case the user cannot establish a data connection via the Wi-Fi access points. 3G cannot be guaranteed in rooms which blocks the signals of mobile telecommunication. In this occasion, it is likely that Wi-Fi signals cannot reached either.

Further recommendations are suggested for the user interface. Enhancements not related maps, but to the input and output, can be done in certain ways. A few are named below:

1. Only a departure and destination can be selected, but it is possible to include via waypoints, or to exclude via waypoints. For instance, a user would like to go from one point in the building, to another point, but the user would like to have a copy machine on his way, and the user would like to avoid a certain section, since there are ongoing refurbishments in that particular part of the building.
2. The inclusion of routes for impaired users, such as wheelchair users. In the database, it has been taken into account which routes are navigable for wheelchairs, as for each edge, information is stored whether a ramp is available, or where the elevators are located. In the prototype, there has been room left for this, as in the Settings screen this has already been created.
3. The inclusion of text-to-speech. Android has a text-to-speech engine, in which the user can hear the directions the application provides. This might help the user, when the user prefers to have his smartphone socketed in his pocket for instance.

There are also limitations and recommendations regarding privacy and legal issues, which are explained in the next chapter.

19.3. Regarding the localization methodologies

In this project, two localization methodologies have been proposed, in which one has been used: the count method was the preferred method over the least sum of squares due to issues in the way how Android treats weak and bad signals. Aside from that, the number of entries per location node influences the outcome of the least sum of squares, as well as the maximum number of valid records. Further research on both methodologies in thus necessary.

These methods behave different than the map-matching algorithms as discussed in the literature research, since there is no geometry to match with. This assumes that there is a given location, in which both location information and signal strength characteristics can be stored. By combining this information, a node can be created, and eventually, a graph, in which various network algorithms can be applied. This can be further extended by explicitly stating which adjacent rooms belong to which node. This way, geometry is not being used, but rather the connectivity and topology is preserved. The importance of topology must be emphasized in further research: since in route guidance, no left/right can be distinguished, alternative solutions must be found. User awareness plays a role here: route confirmation can also made possible by using logics from the real environment, such as “you are still on route, if you can see the signs of the office room numbers are adding up”.

In addition, MAC addresses themselves are dynamic and can be changed at will. This leads to unstable databases in time: it is recommended that upon up-scaling to larger coverage areas, updates should occur frequently. Aside from that, the signal strength characteristics still remain prone to multipath effects, which results in false location information. This can be countered by effectively using the MACs: for instance: only use MACs that are at the same floor as the fingerprint. This prevents looking for outlying signal strengths, which can be unstable anyway.

The search space in which the signal strengths have to be looked up from, can be changed at will as well. The larger the search space, the more problems are foreseen with matching locations: as the larger it is, the more locations can be made available. Moreover, it is recommended that the locations to be registered should not be too far away from each other, as otherwise, intermediate locations are being detected as somewhere else, while that has not be the case per say. On the other hand, the to be fingerprinted locations should not be too close to each other, since it is then difficult to perceive which location belongs to what. It is noteworthy, that the intention of this project was to map locations and not map each single possible position – there is simply too much differentiation in signal strength to do so.

Further automatizing is possible by developing an application, which is capable of directly registering average signal strengths from MACs, in which location information can be assigned. By repeating this for several spots, and by adding information about which spots are connected in physical space, it is easy to build up a graph.

Finally, the databases are dependent from the device being used. A high-end receiver, such as a laptop provides different outcomes than using the smartphone directly for both recording as receiving. A clear disadvantage however, is that multiple databases are needed: each database that should be adapted to the quality of the Wi-Fi receiver.

20. Additional remarks on privacy and legal issues

The application faces multiple issues regarding privacy and legal aspects, which are addressed below. In this chapter a MAC (Media Access Control) refers to the address from a transmittable access point (routers), as non-transmittable devices, such as computers also contain MACs.

20.1. Personal information

Originally, the intention was to include personal information in the application. This is related to the available cataloguing options in the application. In the case a user is looking for the office of a certain person, the user can search for the name of the person, in which the proper room is being given. Thus, the name of a staff member was available in combination with the room. However, this implies that a significant amount of personal information has to come with the application, resulting in an application which can be data intensive, and is vulnerable to hacking as well. Yet, if not too much effort can be done to retrieve personal information via one information channel (such as the office rooms of personnel, which can be browsed by a corporate website), it is probably not too sensitive, hence, it can be used in the application.

20.2. MAC as personal information

Late April 2011, Google was fined €250,000 by a Dutch lawsuit, as Google deliberately recorded all MAC addresses they detected during their coverage of Google Street View (NRC, 2011). Recording the MAC address (the Base SSID, or BSSID) itself is not punishable by law, as long as personal information is not combined with it, according to the College Bescherming Persoonsgegevens (CBP), a quasi-non-governmental organization in the Netherlands, which concerns about the protection of personal data and privacy. However, Google combined the MAC address with a WGS84 Coordinate, along with street names, postal codes, places and more information, in which then the MAC address itself turns into personal data (CBP, 2010; CBP, 2011). Thus a MAC address can be considered as proprietary data. This resulted in a forced deletion of the acquired information, as demanded by the CBP. The law (WBP Article 8.f) is roughly translated (CBP, 2011):

Data about persons may only be processed when data processing is crucial for the (representation of) the justifiable interest for the responsible person, institute or third member to whom the data will be issued, unless the interest or fundamental rights and freedom of the involved person prevails, with special attention regarding the right to the private life.

This issue is strongly related with the application, since it utilizes MAC information, which is being correlated with the received signal strengths. To avoid lawsuits, the application should not be coupled with other personal information.

20.3. Coverage

Correlated with the MAC as personal information depicted and personal information in general in the previous sections, the coverage area itself is prone to privacy and legal issues as well. Since MACs are proprietary (as they can be bought for a specific individual or group), it is reasonable to limit the coverage of a service deployment to public buildings, such as hospitals, conference centres, shopping malls and governmental buildings. This will reduce the amount of personal and sensitive information. Finally, it is most unlikely a user is trying to navigate through very small and refined rooms such as in private objects as houses and apartments. As required by the CBP, upon site surveying, the persons/institutions who own or work with their MAC, should be informed their MAC is being used for location information.

References

Albrecht-Buehler, C., B. Watson & D.A. Shamma (2005), Visualizing Live Text Streams Using Motion and Temporal Pooling. IEEE Computer Society, pp. 52- 59.

Android Developers (2011), Developers Guide - Using Databases. Retrieved [August 18 2011] http://developer.android.com/guide/topics/data/data-storage.html#db.

BIPM, Bureau International des Poids et Mesures (2008), International vocabulary of metrology – Basic and general concepts and associated terms (VIM). Retrieved [February 24 2011] http://www.bipm.org/utils/common/documents/jcgm/JCGM_200_2008.pdf.

Boguslawski, P, C.M. Gold & H. Ledoux (2011), Modelling and analysing 3D buildings with a primal/dual data structure. ISPRS Journal of Photogrammetry and Remote Sensing (66), pp. 188-197.

Bowditch, N. (2002), The American Practical Navigator – An Epitome Of Navigation. 2002 Bicentennial Edition. Bethesda, MD: National Imagery and Mapping Agency.

Chiou, Y., C. Wang, S. Yeh & M. Su (2009), Design of an adaptive positioning system based on WiFi radio signals. Computer Communications 32, pp. 1245-1254.

CBP, College Bescherming Persoonsgegevens (2010), Definitieve bevindingen. Onderzoek CBP naar de verzameling van Wifi-gegevens met Street View auto's door Google. Z2010-00582, openbare versie. Retrieved [April 19 2011]:
http://www.cbpweb.nl/downloads_rapporten/rap_2011_google.pdf

CBP, College Bescherming Persoonsgegevens (2011), Last onder dwangsom / opdracht tot vernietiging payload data. Retrieved [April 19 2011]:
http://www.cbpweb.nl/downloads_pb/pb_20110419_brief_google_lod.pdf

Dawes, B. & K. Chin (2011), A comparison of deterministic and probabilistic methods for indoor localization. The Journal of Systems and Software 84, pp. 442-451.

De Koning, M. (2010), Quality assessment of GSM positioning. GSM fingerprinting versus Cell-ID positioning. MSc Thesis made available upon request.

De La Roche, G., P. Flipo, Z. Lai, G. Villemaud, J. Zhang & J. Gorce (2010), Implementation and validation of a new combined model for outdoor to indoor radio coverage predictions. Journal on Wireless Communications and Networking, pp. 1-9.

Döllner, J., B. Hagedorn & S. Schmidt (2007), An approach towards Semantics-Based Navigation in 3D City Models on Mobile Devices. Lecture Notes in Geoinformation & Cartography 2 (4), pp. 357-368.

Dovis, F., R. Lesca, G. Boiero & G. Ghinamo (2010), A test-bed implementation of an acquisition system for indoor positioning. GPS Solutions 14, pp. 241-253.

ETSI (European Telecommunication Standard Institute) (2010), On-demand Base Maps on the Web generalized according to User Profiles. Internal document submitted.

Foerster, T. (2010), Web-based architecture for on-demand maps – integrating meaningful generalization processing. Retrieved [August 18 2011] http://www.itc.nl/library/papers_2010/phd/foerster.pdf

Haenselmann, T. (2005), Lecture on Sensor Networks. Retrieved [February 24 2011] http://www.slideshare.net/netfet/localization-presentation.

Hsu, C.C. (2011), Factors affecting webpage's visual interface design and style. Procedia Computer Science 3, pp. 1315-1320.

Ibrahim, A. & D. Ibrahim (2010), Real-time GPS based outdoor WiFi localization system with map display. Advances in Engineering Software 41, pp. 1080-1086.

IEEE (2011), IEEE Standard for Information technology — Telecommunications and information exchange between systems — Local and metropolitan area networks — Specific requirements — Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Retrieved [March 1 2011] http://standards.ieee.org/about/get/802/802.11.html.

Intini, A.L. (2000), Orthogonal Frequency Division Multiplexing for Wireless Networks - Standard IEEE 802.11a. Retrieved [March 16 2011] http://www.create.ucsb.edu/ATON/01.01/OFDM.pdf.

Jan, S., L. Hsu & W. Tsai (2010), Development of an Indoor Location Based Service Test Bed and Geographic Information System with a Wireless Sensor Network. Sensors 10, pp. 2957-2974.

Kessler, B. (2006), Direct Sequence Spread Spectrum and You. Presentation Retrieved [March 14 2011] http://www.eecs.tufts.edu/~bkesslr/EE194SDR/Final/DSSS.ppt.

Klukas, R., O. Julien, L. Dong, E. Cannon & G. Lachapelle (2004), Effects on building materials on UHF ranging signals. GPS Solutions 8 (1), pp. 1-8.

Kolodziej, K. & J. Hjelm (2006), Local Positioning Systems. LBS Applications and Services. Boca Raton, FL, USA: CRC Press – Taylor & Francis Group.

Kraak, M.J. & F.J. Ormeling (2003), Cartography – Visualization of Geospatial Data. Second edition. Harlow, UK: Pearson Education Limited.

Le, M.H.V., D. Saragas & N. Webb (2009), Indoor Navigation System for Handheld Devices. A Major Qualifying Project Report. Retrieved [May 20 2011] http://www.wpi.edu/Pubs/E-project/Available/E-project-102209-164024/unrestricted/Indoor_Navigation_System_for_Handheld_Devices.pdf.

Lee, K.-C. (2007), Localization Systems Using Signal Strength Fingerprinting. Retrieved [July 11 2011] https://circle.ubc.ca/bitstream/handle/2429/28750/ubc_2010_fall_lee_kungchung.pdf.

Lim, H. L. Kung, J.C. Hou & H. Luo (2010), Zero-configuration indoor localization over IEEE 802.11 wireless infrastructure. Wireless Networks 16, pp. 405-420.

Lu, F. & P. Lai (2006), A Shortest Path Algorithm Based On Limited Search Heuristics. Lecture Notes on Computer Science (3967), pp. 487-497.

Manodham, T., L. Loyola & T. Miki (2008), A Novel Wireless Positioning System for Seamless Internet Connectivity based on the WLAN Infrastructure. Wireless Personal Communications 44, pp. 295-309.

Mehmood, H., N.K. Tripathi & T. Tipdecho (2010), Indoor Positioning System Using Artificial Neural Network. Journal of Computer Science 6 (10), pp. 1219-1225.

Milioris, D., L. Kriara, A. Papakonstantinou, G. Tzagkarakis, P. Tsakalides & M. Papadopouli (2010), Empiral Evaluation of Signal-Strength Fingerprint Positioning in Wireless LANs. Publication from Institute of Computer Science. Retrieved [July 11 2011] http://www.ics.forth.gr/mobile/publications/mswim2010.pdf.

Nagi, R.S. (2004), Cartographic visualization for mobile applications. Thesis. Retrieved [March 11 2011] http://www.itc.nl/library/papers_2004/msc/gfm/nagi.pdf.

NRC (2011), Dwangsom voor Google om verzamelen wifi-gegevens. News Article Retrieved [April 19 2011]: http://www.nrc.nl/nieuws/2011/04/19/dwangsom-voor-google-om-verzamelen-wifi-gegevens/

OGC (Open Geospatial Consortium) (2008), Open GIS Location Services (OpenLS): Core Services. Retrieved [January 29 2011] http://www.opengeospatial.org.

OGC (Open Geospatial Consortium) (2008), Open GIS Location Services (OpenLS): Part 6-Navigation Service. Retrieved [March 10 2011] http://www.opengeospatial.org.

OGC (Open Geospatial Consortium) (2010), Requirements and Space-Event Modeling for Indoor Navigation. How to simultaneously address route planning, multiple localization methods, navigation contexts and different locomotion types. Retrieved [January 11 2011] http://www.opengeospatial.org.

Reichenbacher, T. (2005), Chapter 10: Adaptive egocentric maps for mobile users. In: L. Meng, A. Zipf & T. Reichenbacher (2005), Map-Based Mobile Services. Theories, Methods and Implementations, pp. 141-158. Springer Berlin: Heidelberg.

Roos, T., P. Myllymäki, H. Tirri, P. Misikangas & J. Sievänen (2002), A probabilistic Appraoch to WLAN User Location Estimation. International Journal of Wireless Information Networks 9 (3), pp. 155-163.

Roto, V. & A. Kaikkonen (2003), Perception of Narrow Web Pages on a Mobile Phone. Proceedings of the 19th International Symposium on Human Factors in Telecommunication (HFT 2003), pp. 205-212.

Sarjakoski, L.T., T. Koviula & T. Sarjakoski (2007), A Knowledge-Based Map Adaptation Approach for Mobile Map Services. Lecture Notes in Geoinformation & Cartography 2 (3), pp. 247-264.

Shum, K.C.Y, Q.J. Cheng, J.K.Y Ng & D. Ng (2011), A Signal Strength based Location Estimation Algorithm with a Wireless Network. 2011 International Conference on Advanced Information Networking and Applications, pp. 509-516.

Sturgeon, T. et al. (2011), Wi-Fi Virtual Laboratory. Retrieved [March 11 2011] http://wifi.cs.st-andrews.ac.uk/wififrame.html.

Swangmuang, N. & P. Krishnamurthy (2008), An effective location fingerprint model for wireless indoor localization. Pervasive and Mobile Computing 4, pp. 836-850.

Usher, J.M. & L. Strawderman (2010), Simulation operational behaviors of pedestrian navigation. Computer & Industrial Engineering (59), pp. 736-747.

Vogel, L. (2009), Dijkstra's Shortest Path Algorithm in Java. Retrieved [May 1 2011] http://www.vogella.de/articles/JavaAlgorithmsDijkstra/index.html

Wang, A.H. & C.H. Chen (2003), Effects of screen type, Chinese typography, text/background color combination, speed, and jump length for VDT leading display on users' reading performance. International Journal of Industrial Ergonomics 31, pp. 249-261.

Wi-Fi Alliance (2009), Wi-Fi Certified n: Longer-Range, Faster throughput, Multimedia-Grade Wi-Fi Networks. Retrieved [March 1 2011] http://www.wi-fi.org/register.php?file=wp_Wi-Fi_CERTIFIED_n_Industry.pdf

Winter (2002), Modeling Costs of Turns in Route Planning. GeoInformatica 6 (4), pp. 345-361.

Woo *et al.* (2011), Application of Wi-Fi-based indoor positioning system for labor tracking at construction sites: A case study in Guangzhou MTR. Automation in construction 20, pp. 3-13.

Wu, Z., Q. Zeng & X. Hu (2009), Mining Personalized User Profile Based on Interesting Points and Interesting Vectors. Information Technology Journal 8 (6), pp. 830-838.

Yim, J. (2008), Introducing a decision tree-based indoor positioning technique. Expert Systems with Applications 34, pp. 1296-1302.

Yim, J., S. Jeong, K. Gwon & J. Joo (2010), Improvement of Kalman filters for WLAN based indoor tracking. Expert Systems with Applications 37, pp. 426-433.

Zhang, D. *et al.* (2010), Localization Technologies for Indoor Human Tracking. Paper submitted to the 5th International Conference on Future Information Technology (FutureTech), May 2010, Busan, Korea. Retrieved [January 26 2011] http://arxiv.org/ftp/arxiv/papers/1003/1003.1833.pdf.

Annexes

A. Tables and scripts

A.1. Table of overview possible sensor technologies for location based services

Technique	Range	Accuracy	Remarks	Allocation methods
(Assisted) Global Positioning System	*Anywhere outdoor*	6.0 m - 12.0 m	→ Satellite based positioning + Low barrier entry - Energy consumptive (Assisted GPS in a lesser extent) - Slow computation and processing time - Very susceptible to reflectance and multi-paths	Trilateration
Global System for Mobile Communication (GSM) / Universal Mobile Telecommunication System (UMTS)	≈ 35.0 km	*Cell-based*	→ Standard data- and telephone communication radio waves. + Globally available - Cell-based accuracy	Cell-ID Signal Strength
Infrared (IR)	0.7 m – 2.5 m	*Dependent on range of application*	→ Device tagging with ID's - Short range of detection limits infrastructure - No penetration of materials / multipath - Line of sight - Signal can be disturbed easily	Cell-ID
Infrared Data Association (IrDA)	≈ 2.5 m	1.0 m – 2.0 m	→ Same method as IR, yet with higher speed	Cell-ID
Radio Frequency Identification (RFID)			→ Radio waves exchange between reader and tag, using a transmitter, transceiver with decoder and unique information. + Real-time location systems + High-speed response time + read/write capabilities - No communication network - No positioning information - Manual programming	Cell-ID
Active RFID	< ≈ 100 m		→ Tags contain internal power source, signalling upon entering range (*toll boots*)	
Passive RFID	1.5 m – 2.0 m		→ No internal power source, only reading/writing upon entering range/connection (*smartcards)*	

Technique	Range	Accuracy	Remarks	Allocation methods
Ultrasound	*Dependent on environment*	3.0 cm – 1.0 m	→ Pulse (radar) emittance - No penetration of materials / multipath - Line of sight - Excessive manual intervention - Bad reception with intervening noises – extremely sensitive to environment	Trilateration
Bluetooth	< ≈ 100 m	10 m – 20m	→ Frequency hopping for connecting devices and creating Personal Area Networks (PAN), with high security + High speed data transfer - Explicit links between devices required - Its mobility also limits positioning and topology	Trilateration Signal Strength
Ultra Wide Band (UWB)	≈ 50 m (*with reduced accuracy, normal: chips about 9 m, due to cap of power amount)*	15 cm – 4 m	→ Extremely short emittance of pulses + Multipath immunity + Inherent precision for TOA/TDOA + Low power + High speed data (nearly 10xWi-Fi) - Large uncertainty about health effects - Not everywhere legal - Economically expensive	Trilateration Signal Strength
IEEE 802.11 (Wi-Fi)	≈ 32 m (indoor) ≈ 95 m (outdoor)	1 m – 5 m	→ Radio waves transfer at the 2.4 GHz band + Large scale available over the world + Economical viable - High power consumption - Security can be weak - Slightly multipath susceptible *+ 802.11n at 5 GHz band, range indoor 91m, outdoor 182m*	Trilateration Signal Strength

Table AP. 1. Extended overview of sensor techniques used for allocation of receiver.

Table based on Manodham et al. (2008) and Kolodziej & Hjelm (2006) and Lim et al. (2010).

A.2. Navigation flows

The OGC (2008b) has listed a set of 30 possible cases (flows) which determines or influences the navigation. They are listed below:

Flow	Flow description	*Remark*
Main flow		
1	User specifies routing preferences.	
2	User requests a route from the client a route from origin to destination.	
3	Client requests current position.	*In this project detection of fingerprint*
4	Client returns network link(s) (and possibly locations along) as origin.	
5	User specifies destination.	
6	Client notes the returned network link(s) (and possibly locations along) as the destination.	*Geocoding service*
7	Client requests navigation service to provide a route from origin to destination.	
8	Client informs user and guidance function route has been generated.	
9	Client displays route.	
Extension: navigation request detour		
10	User requests detour re-route after starting route guidance	
11	User requests traffic information and client makes the request of navigation or traffic server	*In general not applicable for indoor navigation, unless in crowded buildings, such as supermarkets*
12	User supplies detour information (avoidances)	
13	Client request navigation or route service to provide a new route based on detour	
14	Navigation client displays new route on the map	
15	User accepts new route	
16	Client informs route guidance function a detour route has been generated	
Extension: rerouting with new waypoints		
17	While route guidance is underway, user sets additional waypoint(s)	*Can be done to include certain points to the route*
18	Client returns location(s) as additional waypoint(s)	
19	Client requests navigation (or route) service to provide a new route, from current location to original destination, including the new waypoints not yet visited	
20	Client informs user and route guidance function a way-pointed route has been generated	
21	Client displays the new route	
Extension: recovery reroute		
22	User travels off current guided route	

Flow	Flow description	*Remark*
23	Client detects user is off-route	*Assumes that fingerprints does no longer correlate*
24	Client request navigation service to provide a recovery route from the current user position back onto the current route	
25	Client informs user and route guidance recovery route has been generated	
26	Client displays the recovery route	
Extension: change of routing preference		
27	User requests change of routing preferences	
28	Client provides list of preferences supported	
29	User selects items and submits	
30	Client verifies selection and establish them as new current settings, and to re-plan route if necessary	

Table AP. 2. OGC Navigation flows

Source: OGC (2008b)

A.3. Database contents

The relations of the tables can be found in Figure 26. The fields are explained in the tables below.

Table: table_location (nodes)		
Field	**Type**	**Description**
Fid	Number	Fingerprint location ID
Type_exit	Boolean	Exit available at id?
Type_room	Boolean	Room(s) available at id?
Type_junction	Boolean	Junction(s) available at id?
Type_stairs	Boolean	Staircase(s) available at id?
Type_elevator	Boolean	Elevator(s) available at id?
Type_aisle	Boolean	Aisle(s) available at id?
Building	String	Name of building fingerprint measured
Address	String	Address of building fingerprint measured
Place	String	Place of building fingerprint measured
Floor	String	Floor of building fingerprint measured
Wing	String	Wing of building fingerprint measured
Useraccess	Boolean	Restricted or unrestricted at point measured
Locprop	String	Objects to be seen at fingerprint

Table AP. 3. Database: fingerprint locations

Table: table_connectivity (edges)		
Field	**Type**	**Description**
cid	Number	Number of connectivity link ID
Fid_link_01	Number	Link between FID (1)
Fid_link_02	Number	Link between FID (2)
Length	Number	Length of link in metres
Direction	String	Movement of link in reality: straight, up, down
Restricted	Boolean	Restricted or unrestricted movement
Stairs	Boolean	Movement along stairs possible?
Elevator	Boolean	Movement along elevator possible?
Ramp	Boolean	Movement along ramp possible?
Door	Boolean	Movement through door possible?
Objects_left	String	Description of objects left of the node
Objects_right	String	Description of objects right of the node

Table AP. 4. Database: connectivity

Table: table_rooms (directory)		
Field	**Type**	**Description**
Rid	Number	Number of room ID
Fid	Number	Proximity of Fingerprint ID
Room_name	String	Name of room (browsable)
Wing	String	Room in wing of building
Floor	String	Room on floor of building
Building	String	Building name
Department	String	Department present in room
Office	String	Office present in room
Room_type	String	Type of room (office, kitchen, toilet etc.)
Restricted	Boolean	Restricted or unrestricted to staff
Orientation	String	Orientation regarding topology

Table AP. 5. Database: room information

Table: table_POI (directory, unused)		
Field	**Type**	**Description**
Poid	Number	Number of POI
Poigroup	Number	Category of POI
Poisubgroup	Number	Sub category of POI
Fid	Number	POI located at Fingerprint ID
Poispecification	String	Comments on POI

Table AP. 6. Database: POI information

A.4. Database samples

The tables below provide examples of the SQLite database in the form of statements in XML files. The full files can be found in the attached xml files, to be found under `Indoor Nav/res/raw`.

```
<sql>
       <statement>
               CREATE TABLE IF NOT EXISTS table_location (
               fid INTEGER PRIMARY KEY,
               type_exit BOOLEAN,
               type_room BOOLEAN,
               type_junction BOOLEAN,
               type_stairs BOOLEAN,
               type_elevator BOOLEAN,
               type_aisle BOOLEAN,
               building VARCHAR(50),
               address VARCHAR(100),
               place VARCHAR(100),
               floor INTEGER,
               wing VARCHAR(50),
               useraccess VARCHAR(50),
               locprop_01 VARCHAR(200),
               locprop_02 VARCHAR(200),
               locprop_03 VARCHAR(200))
</statement>
<statement>INSERT INTO table_location VALUES(2001,1,0,0,0,0,0,'OTB','Jaffalaan
9','Delft',0,2,'Unrestricted','Outside Entrance','','');</statement>
<statement>INSERT INTO table_location VALUES(2002,1,0,1,0,0,1,'OTB','Jaffalaan
9','Delft',0,2,'Unrestricted','Entrance','','');</statement>
<statement>INSERT INTO table_location VALUES(2003,0,0,1,0,0,1,'OTB','Jaffalaan
9','Delft',0,2,'Unrestricted','Path to Students Affairs and Back Exit','','');</statement>
<statement>INSERT INTO table_location VALUES(2004,0,0,0,0,0,1,'OTB','Jaffalaan
9','Delft',0,2,'Unrestricted','Hall to back exit','','');</statement>
```

Box AP. 1. Sample database for table_location

```
<sql>
<statement>
               CREATE TABLE IF NOT EXISTS table_mac (
               fid INTEGER,
               mac VARCHAR(20),
               rssi_avg INTEGER,
               CONSTRAINT table_mac
               PRIMARY KEY (fid, mac)
               )
</statement>
<statement>INSERT INTO table_mac VALUES(2001,'00:1A:A2:C0:0D:70',-76);</statement>
<statement>INSERT INTO table_mac VALUES(2001,'00:1A:A2:C0:0D:72',-78);</statement>
<statement>INSERT INTO table_mac VALUES(2001,'00:1A:A2:C0:0D:74',-76);</statement>
<statement>INSERT INTO table_mac VALUES(2001,'00:1A:A2:C0:0D:75',-77);</statement>
```

Box AP. 2. Sample database for table_mac

```
<sql>
	<statement>
		CREATE TABLE IF NOT EXISTS table_connectivity (
		cid INTEGER
		PRIMARY KEY,
		length INTEGER,
		fid_link_01 INTEGER,
		fid_link_02 INTEGER,
		direction VARCHAR(10),
		restricted BOOLEAN,
		stairs BOOLEAN,
		elevator BOOLEAN,
		ramp BOOLEAN,
		door BOOLEAN,
		objects_left VARCHAR(200),
		objects_right
		VARCHAR(200))
	</statement>
	<statement>INSERT INTO table_connectivity
VALUES(20001,5,2001,2002,'straight',1,0,0,0,1,'','');</statement>
	<statement>INSERT INTO table_connectivity
VALUES(20002,5,2002,2001,'straight',1,0,0,0,1,'','');</statement>
	<statement>INSERT INTO table_connectivity
VALUES(20003,5,2002,2003,'straight',1,0,0,0,0,'','');</statement>
	<statement>INSERT INTO table_connectivity
VALUES(20004,5,2002,2006,'straight',1,0,0,0,0,'','Info Pillar, Sofas');</statement>
```

Box AP. 3. Sample database for table_connectivity

```
<sql>
	<statement>
		CREATE TABLE IF NOT EXISTS table_rooms (
		rid INTEGER PRIMARY
		KEY,
		fid INTEGER,
		room_name VARCHAR(50),
		wing INTEGER,
		floor INTEGER,
		building VARCHAR(50),
		department VARCHAR(50),
		office VARCHAR(50),
		room_type VARCHAR(50),
		restricted BOOLEAN))
</statement>
	<statement>INSERT INTO table_rooms VALUES(2002,2008,'Survey
		Room',2,0,'OTB','','Room','Computer Room',1);</statement>
	<statement>INSERT INTO table_rooms VALUES(2004,2008,'Grote
		Vergaderzaal',2,0,'OTB','','Room','Commission',1);</statement>
	<statement>INSERT INTO table_rooms VALUES(2006,2009,'Anneke van
		Kootenzaal',2,0,'OTB','','Room','Commission',1);</statement>
	<statement>INSERT INTO table_rooms VALUES(2007,2010,'0.070
		(Secretary)',2,0,'OTB','','Work Office','Office',1);</statement>
```

Box AP. 4. Sample database for table_rooms

```
<sql>
<statement>
		CREATE TABLE IF NOT EXISTS table_poi (
		poid INTEGER PRIMARY KEY,
		poigroup VARCHAR(50),
		poisubgroup VARCHAR(50),
		fid INTEGER,
		poispecification VARCHAR(50))
</statement>
<statement>INSERT INTO table_poi VALUES(99901,'Commission Rooms','',2009,'Anneke van
Cootenzaal');</statement>
<statement>INSERT INTO table_poi VALUES(99902,'Information','',3006,'Bulletin
Board');</statement>
<statement>INSERT INTO table_poi VALUES(99903,'Information','',3106,'Bulletin
Board');</statement>
<statement>INSERT INTO table_poi VALUES(99904,'Information','',3112,'Bulletin
Board');</statement>
```

Box AP. 5. Sample database for table_poi (unused)

A.5. Codes for location determination

The code below contains the location determination as described in section 16.3.4. The numbers to the left represents the number of lines, which are being described in the main thesis.

```
Public class MainScreen extends Activity implements OnClickListener {
        String departure;
        String destination;
        private SQLiteDatabase db;

        @Override
        public void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.main);

                AutoCompleteTextView dep1 = (AutoCompleteTextView)
findViewById(R.id.inputDeparture);
                RoomList roomList = new RoomList(this);
                ArrayList<String> items1 = roomList.getAllRooms();

                Toast.makeText(this, "Amount of rooms loaded: " +
items1.size(),Toast.LENGTH_SHORT).show();

                                ArrayAdapter<String> adapter1 = new
ArrayAdapter<String>(this,
                                R.layout.list_item, items1);
                dep1.setAdapter(adapter1);

                AutoCompleteTextView des1 = (AutoCompleteTextView)
findViewById(R.id.inputDestination);
                ArrayAdapter<String> adapter2 = new ArrayAdapter<String>(this,
                                R.layout.list_item, items1);
                des1.setAdapter(adapter2);

                Button StartNavigationButton = (Button)
findViewById(R.id.buttonNavigationStart);
                StartNavigationButton.setOnClickListener(this);
                Button StartSettingsButton = (Button) findViewById(R.id.buttonSettings);
                StartSettingsButton.setOnClickListener(this);

                GenericService gs = new GenericService();
                // This line initiates an object (gs) for Wi-Fi service

                gs.SetActivitiy(this);
                // This line lets the object know about your activity

                gs.AddAccessPoint("00:1a:a2:c0:0d:70","00:1a:a2:c0:0d:70");
                gs.AddAccessPoint("00:1a:a2:c0:0d:72","00:1a:a2:c0:0d:72");

                //Add AP to listen, you can add multiple APs by using this method many
times
                gs.AddRule("00:1a:a2:c0:0d:70","00:1a:a2:c0:0d:70", Rule.RULE_TYPE_WIFI, -
90, -35, 1, null);
                gs.AddRule("00:1a:a2:c0:0d:72","00:1a:a2:c0:0d:72", Rule.RULE_TYPE_WIFI, -
90, -35, 1, null);

                db = (new DatabaseHelper(this).getWritableDatabase());
                // Set rules to be informed when given criteria is satisfied by using:
                ListenWifi listener= new ListenWifi(db, handler,this);

                // Create listeners to catch the events thrown by the service...
                gs.AddWifiListener(listener);
                // ...and let the object know where your listener is
                gs.start();
                //and start to wait events
        }

        public void onClick(View v) {
                if (v.getId() == R.id.buttonNavigationStart) {
                        EditText inputDeparture = (EditText)
findViewById(R.id.inputDeparture);
                        EditText inputDestination = (EditText)
findViewById(R.id.inputDestination);

                        if (inputDeparture.getText().length() == 0) {
                                Toast.makeText(this, "Please input departure",
```

```
                        Toast.LENGTH_SHORT).show();
                return;
            } else if (inputDestination.getText().length() == 0) {
                Toast.makeText(this, "Please input destination",
                        Toast.LENGTH_SHORT).show();
                return;

            } else if (inputDeparture == inputDestination) {
                Toast.makeText(this, "Departure and Destination equals",
                        Toast.LENGTH_SHORT).show();
                return;
            }

            Intent n = new Intent(this, Navigation.class);
            n.putExtra("departure", inputDeparture.getText().toString());
            n.putExtra("destination", inputDestination.getText().toString());
            startActivity(n);
        } else if (v.getId() == R.id.buttonSettings) {
            Intent s = new Intent(this, Settings.class);
            startActivity(s);
        }
    }

    @Override
protected void onDestroy() {
    db.close();
    super.onDestroy();
}

//This is the place where we handle messages coming to the GUI thread
private final Handler handler = new Handler(){
    @Override
    public void handleMessage(Message msg) {
        //gets parameter is the type of the message
        String comingMsg = (String)msg.getData().get("2");

        if(comingMsg == null){
            comingMsg = (String)msg.getData().get("1");

            //if there is such a message
            if(comingMsg != null){
                //try to separate the message again
                String fid, floor, wing, building, address, place, locprop;

                //separation process
                String delims = "[;]";

                //split the message into array of strings
            String[] data = comingMsg.split(delims);

            fid = data[0];
            floor = data[1];
            wing = data[2];
            building = data[3];
            address = data[4];
            place = data[5];
            locprop = data[6];

            //Use this variables to set the text now
            TextView textBuilding = (TextView)
findViewById(R.id.currentBuilding);
            textBuilding.setText("At: " + building + ", " + address + ", " +
place);

            TextView textFid = (TextView) findViewById(R.id.currentType);
            textFid.setText("Near: " + locprop + " (Node: " + fid + ")");

            TextView textWing = (TextView) findViewById(R.id.currentWing);
            textWing.setText("Wing: " + wing + ", Floor: " + floor);
            }
            else{
                //No fid is matched!
            }
        }

        super.handleMessage(msg);
    }
```

```
};


class ListenWifi implements WifiListener{
        public static final String MSG_WIFIFOUND = "WifiFound";
        public static final String MSG_WIFILOST = "WifiLost";
        public static final String MSG_WIFIRULE = "WifiRule";
        public static final String MSG_WIFILEVELCHANGED = "WifiChange";
        SQLiteDatabase db;
        Handler handler;
        Activity activity;

        ListenWifi(SQLiteDatabase db, Handler handler, Activity activity){
                this.db = db;
                this.handler = handler;
                this.activity = activity;
        }

        private void SendMessage(String type, String str){
                Bundle b = new Bundle();
                b.putString(type, str);
                Message msg = new Message();
                msg.setData(b);
                handler.sendMessage(msg);
        }

@Override
public void OnStateChange(WifiEvent event){
}

@Override
public void OnAPFound(WifiEvent event) {
        event.GetAccessPoint().getName();
}

@Override
public void OnAPLost(WifiEvent event) {
}

@Override
public void OnRuleAchieved(RuleEvent event) {
        WifiManager wifimanager =
(WifiManager)activity.getSystemService(Context.WIFI_SERVICE);
        wifimanager.startScan();
        List<ScanResult> ls = wifimanager.getScanResults();

        //A loop to customize the query according to the measured values
        String queryHead = "Select fid, count(*) from table_mac where ";
        String queryTail = " group by fid";

        //Vector<String> vQuery = new Vector<String>();
        int counter = 0;
        String queryMid = "";

        if(ls.size() > 0){
                for(ScanResult sr: ls){
                        //Prepare
                        String mac = sr.BSSID;
                        int SS = sr.level;
                        if(counter == 0){
                                queryMid = "(mac  = '"+mac+"' and rssi_avg < "+(SS+4)+" and
rssi_avg > "+(SS-4)+")"; // First MAC with upper and lower
                        }
                        else{
                                queryMid += " or (mac  = '"+mac+"' and rssi_avg <
"+(SS+4)+" and rssi_avg > "+(SS-4)+")"; // Subsequent MAC with upper and lower
                        }
                        counter++;
                } //End of loop scan results

                String finalQuery = queryHead + queryMid + queryTail; // Query buildup

                int max = 0;
                int selectedFID = 0;
                Cursor cursor = db.rawQuery(finalQuery, null);
                if(cursor.getCount() > 0){
                        while(cursor.moveToNext()){
```

```
                        int fid = cursor.getInt(0);
                        int count = cursor.getInt(1);

                        if(count > max){
                            max = count;
                            selectedFID = fid;
                        }//If some records are equal, top one is selected
                    }
                }

                if(selectedFID > 0){
                    String query = "select fid, floor, wing, building, address, place,
locprop from table_location where fid = " + selectedFID ;

                    //New cursor is defined to handle new query results
                    Cursor cursorFID = db.rawQuery(query, null);
                    if(cursorFID.getCount() > 0){
                        cursorFID.moveToNext();

                    String msg = cursorFID.getString(0)+ ";" +cursorFID.getString(1) +
";" + cursorFID.getString(2) + ";" + cursorFID.getString(3) + ";" +
cursorFID.getString(4) + ";" + cursorFID.getString(5) + ";" + cursorFID.getString(6);
                    SendMessage("1", msg);
                }
                else{
                    String msg = "!";
                    SendMessage("2", msg);
                }
                }
            }
}

@Override
public void OnAPLevelChanged(WifiEvent event) {

}
}// end of listener class
```

Box AP. 6. Overview of code for location determination (MainScreen.java and Navigation.java)

Source: M. Yildirim (2011)

A.6. Codes for navigation algorithm (Dijkstra's algorithm)

The tables/codes below provide an overview on how the scripts work. Nodes and edges are being loaded via Vertex.java and Edge.java, in which a graph will be constructed in Graph.java. Finally, those three inputs are being used in DijkstraAlgorithm.java, in which the calculation of the shortest path is being made.

```
Public class DijkstraAlgorithm {
	private final List<Vertex> nodes;
	private final List<Edge> edges;
	private Set<Vertex> settledNodes;
	private Set<Vertex> unSettledNodes;
	private Map<Vertex, Vertex> predecessors;
	private Map<Vertex, Integer> distance;

	public DijkstraAlgorithm(Graph graph) {
		// create copy of array
		this.nodes = new ArrayList<Vertex>(graph.getVertices());
		this.edges = new ArrayList<Edge>(graph.getEdges());
	}

	public void execute(Vertex fid_link_01) {
		settledNodes = new HashSet<Vertex>();
		unSettledNodes = new HashSet<Vertex>();
		distance = new HashMap<Vertex, Integer>();
		predecessors = new HashMap<Vertex, Vertex>();
		distance.put(fid_link_01, 0);
		unSettledNodes.add(fid_link_01);
		while (unSettledNodes.size() > 0) {
			Vertex node = getMinimum(unSettledNodes);
			settledNodes.add(node);
			unSettledNodes.remove(node);
			findMinimalDistances(node);

			// fid_link_01 = source, fid_link_02 = destination
		}
	}

	private void findMinimalDistances(Vertex node) {
		List<Vertex> adjacentNodes = getNeighbors(node);
		for (Vertex target : adjacentNodes) {
			if (getShortestDistance(target) > getShortestDistance(node) +
getDistance(node, target)) {
					distance.put(target, getShortestDistance(node) +
getDistance(node, target));
					predecessors.put(target, node);
					unSettledNodes.add(target);
				}
		}
	}

	private int getDistance(Vertex node, Vertex target) {
		for (Edge edge : edges) {
			if (edge.getFid_link_01().equals(node) &&
edge.getFid_link_02().equals(target)) {
				return edge.getLength();
				// Length = weight in this case
				}
			}
		throw new RuntimeException("should not happen");
	}

	private List<Vertex> getNeighbors(Vertex node) {
		List<Vertex> neighbors = new ArrayList<Vertex>();
		for (Edge edge : edges) {
			if (edge.getFid_link_01().equals(node) &&
!isSettled(edge.getFid_link_02())) {
				neighbors.add(edge.getFid_link_02());
			}
		}
		return neighbors;
	}

	private Vertex getMinimum(Set<Vertex> vertices) {
```

```
		Vertex minimum = null;
		for (Vertex vertex : vertices) {
			if (minimum == null) {
				minimum = vertex;
			} else {
				if (getShortestDistance(vertex) < getShortestDistance(minimum))
{
					minimum = vertex;
				}
			}
		}
		return minimum;
	}

	private boolean isSettled(Vertex vertex) {
		return settledNodes.contains(vertex);
	}

	private int getShortestDistance(Vertex fid_link_02) {
		Integer d = distance.get(fid_link_02);
		if (d == null) {
			return Integer.MAX_VALUE;
		} else {
			return d;
		}
	}
	/*
	 * Returns path from source (fid_link_01) to selected target and NULL if no path
	 */
	public LinkedList<Vertex> getPath(Vertex target) {
		LinkedList<Vertex> path = new LinkedList<Vertex>();
		Vertex step = target;
		// Check if a path exists
		if (predecessors.get(step) == null) {
			return null;
		}
		path.add(step);
		while (predecessors.get(step) != null) {
			step = predecessors.get(step);
			path.add(step);
		}
		// Put it into the correct order
		Collections.reverse(path);
		return path;
	}
```

Box AP. 7. Code for DijkstraAlgorithm.java

Edited after: L. Vogel (2009)

```
public class Graph {
	public final List<Vertex> vertices;
	public final List<Edge> edges;

	public Graph(List<Vertex> vertices, List<Edge> edges) {
		this.vertices = vertices;
		this.edges = edges;
	}

	public List<Vertex> getVertices() {
		return vertices;
	}

	public List<Edge> getEdges() {
		return edges;
	}
}
```

Box AP. 8. Code for Graph.java

Edited after: L. Vogel (2009)

```
public class Vertex {
	final private Integer fid;
	final private String room_name;
	final private ArrayList<Room> rooms;

	public Vertex(Integer fid, String room_name, ArrayList<Room> rooms) {
		this.fid = fid;
		this.room_name = room_name;
		this.rooms = rooms;
	}

	public int getFid() {
		return fid;
	}

	public String getRoomName() {
		return room_name;

	}

	public ArrayList<Room> rooms() {
		return this.rooms;
	}

	@Override
	public int hashCode() {
		final int prima = 31;
		int result = 1;
		result = prima * result + ((fid == null) ? 0 : fid.hashCode());
		return result;

	}

	@Override
	public boolean equals(Object obj) {
		if (this == obj)
			return true;
		if (obj == null)
			return false;
		if (getClass() != obj.getClass())
			return false;
		Vertex other = (Vertex) obj;
		if (fid == null) {
			if (other.fid != null)
				return false;
		} else if (!fid.equals(other.fid))
			return false;
		return true;
	}

	@Override
	public String toString() {
		return room_name;
	}
}
```

Box AP. 9. Code for Vertex.java

Edited after: L. Vogel (2009)

```
public class Edge {
	private final int cid;
	private final Vertex fid_link_01;
	private final Vertex fid_link_02;
	private final int length;
	private final String direction;
	private final String restriction;
	private final boolean stairs;
	private final boolean elevator;
	private final boolean ramp;
	private final boolean door;
	private final String object_left;
	private final String object_right;

	public Edge(int cid, Vertex fid_link_01, Vertex fid_link_02, int length, String 
direction, String restriction, boolean stairs, boolean elevator, boolean ramp, boolean door, 
String object_left, String object_right) {
		this.cid = cid;
		this.fid_link_01 = fid_link_01;
		this.fid_link_02 = fid_link_02;
		this.length = length;
		this.direction = direction;
		this.restriction = restriction;
		this.stairs = stairs;
		this.elevator = elevator;
		this.ramp = ramp;
		this.door = door;
		this.object_left = object_left;
		this.object_right = object_right;
	}

	public int getCid() {
		return cid;
	}
	public Vertex getFid_link_01() {
		return fid_link_01;
	}
	public Vertex getFid_link_02() {
		return fid_link_02;
	}
	public int getLength() {
		return length;
	}
	public String getDirection() {
		return direction;
	}
	public String getRestriction() {
		return restriction;
	}
	public boolean getStairs() {
		return stairs;
	}
	public boolean getElevator() {
		return elevator;
	}
	public boolean getRamp() {
		return ramp;
	}
	public boolean getDoor() {
		return door;
	}
	public String getObject_left() {
		return object_left;
	}
	public String getObject_right() {
		return object_right;
	}
	@Override
	public String toString() {
		return fid_link_01 + " " + fid_link_02;
	}
}
```

Box AP. 10. Code for Edge.java

Edited after: L. Vogel (2009)

B. Background literature

B.1. Background information on pedestrian movements and navigation algorithms

Another difference with outdoor navigation, is that (indoor) pedestrian movement is often not variable in speed, making the shortest path the fastest path. Usher and Strawderman (2010) investigated the pedestrian movement, in where they noticed some remarkable properties: (1) people move with an average speed of 1.57 m/s (5.7 km/h) for fair open places and 1.47 m/s (5.3 km/h) for narrow places, depending on the preferred walking speed, and depending on how crowded a place can be; (2) people try to change the angular movement as limited as possible, keeping their route as smooth as possible. This implies that not always the shortest (fastest) path in indoor environments will be followed. In addition, Winter (2002) researched the modelling of turn costs in route planning for the outdoor navigation. This includes the disallowance of turning in some situations, where a modality need to look for alternatives, such as driving to the nearest roundabout, make a U-turn and still drive in the desired direction. Pedestrian movements are in general not restricted to such features, as they freely move from one point to another. Beside physical constraints, a pedestrian is not allowed to turn when the designated space is constrained by authority, such as opening hours. The information above can be used to determine estimated times of arrival.

Figure AP. 1 summarizes the above contents. In building A, it takes 20 units from a pedestrian to walk from one end to another end. If the building was structured as a U-shape (B), then it would take 30 units to walk. If there would be suspended corridor on another floor in this same building (C), one could consider taking the staircases. However, the travel costs for using the staircase, might depend on how the pedestrian perceives such vertical movement. For instance, if the pedestrian does not bother using the stairs, a value of 1 for one vertical movement can be assigned, ending up in a total of 3+1+12+1+3=20 units. If the pedestrian finds it annoying, a value of 5 could be assigned, ending up in a total of 3+5+12+5+3=28 units. In addition, the risk of getting lost (the second comment of Usher & Strawderman (2010), as stated above), might be incorporated as well, with the result that this shortest path might be inconvenient.

A possible inclusion of the exterior to be used for indoor positioning and the shortest paths is recently mentioned by Boguslawski, Gold & Ledoux (2011). Sometimes, it might pay off to leave the building, and enter the same building through another door at another side. This could be the case in Figure AP. 1B if this would be located on the ground floor, and if exits/entrances would exist at the northern edges. One could easily walk from on and to another. If the Wi-Fi access points could reach that area, then it is surely possible to use such outdoor fingerprints in the model.

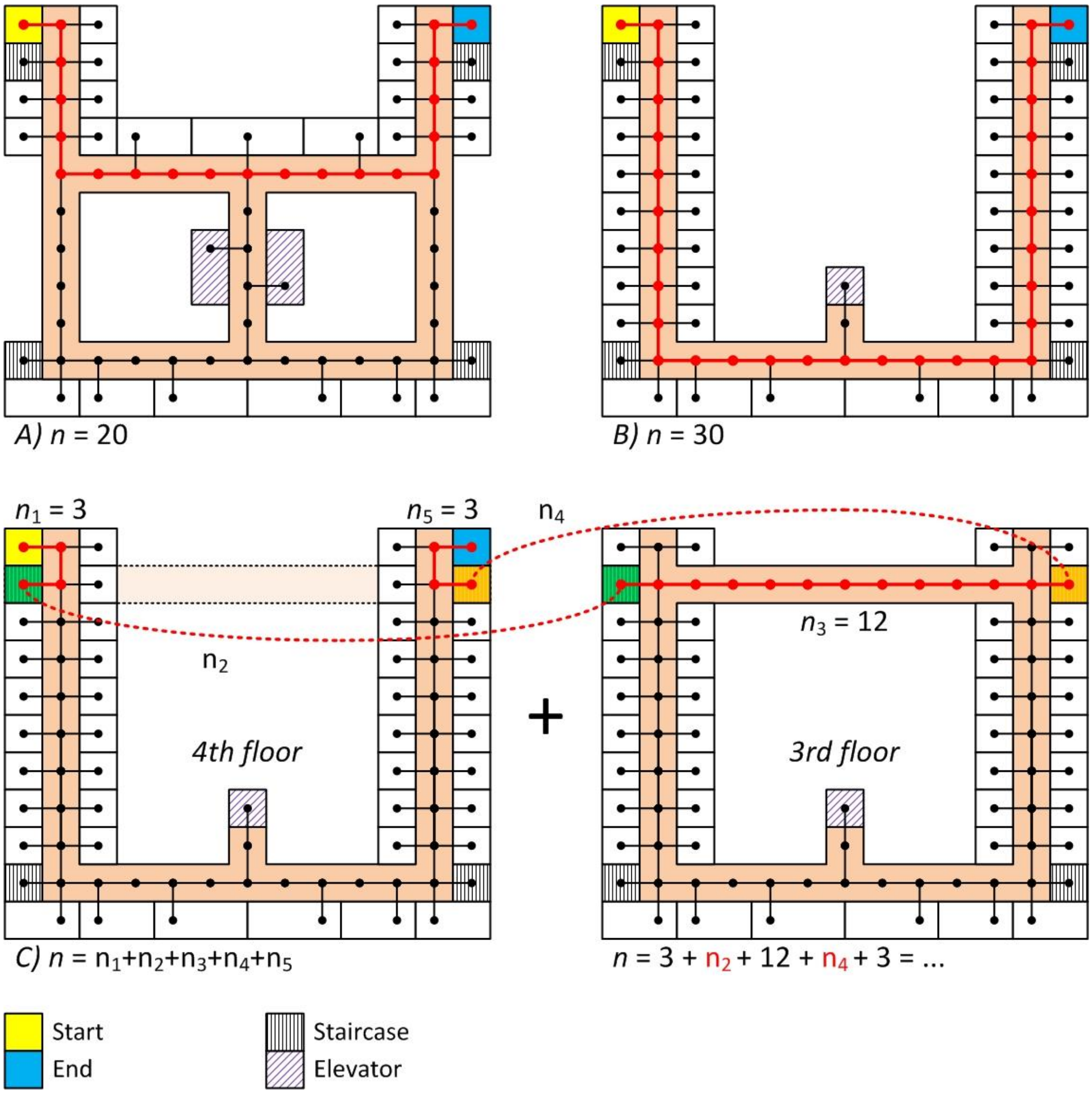


Figure AP. 1. Shortest paths in buildings

B.2. Mobile cartography and the user

The user profile has influence on the visualization and use of maps Sarjakoski et al. (2007), underlines the importance of user adaptive maps. According to them, the final map on mobile phones is dependent on the following parameters:

- Use case, also to be considered for which purpose the map should serve;
- Level of detail, expressed in the resolution of the map;
- Time, the map's time stamp (the up-to-datedness);
- Age, as in the age group of the user;
- Device, related to the capabilities and constraints of the mobile device;
- Centre, as in the centre point of the map;
- Scale, which is the display scale; and
- Position, which is the position of the user on the map.

This assumes there is a strong correlation of the user's knowledge of maps and map usage in general. It also determines which data should be displayed on the map first. Sarjakoski et al. (2007), proposes to use a map specification knowledge base (MSKB) to store all kinds of combinations and possibilities, so the actual map representation can be adjusted accordingly. These will affect the placement and layout of:

- Topographic features to be displayed;
- Points/Areas/Lines of Interest;
- Level of detail;
- Other visualization operators, such as icon placement; and
- Other map visualization, such as colours and line widths.

In fact, these ideas are similar as described by Reichenbacher (2005), who made the following graphic interpretation (Figure AP. 2), on how user, information, situation, activities and the device are interrelated.

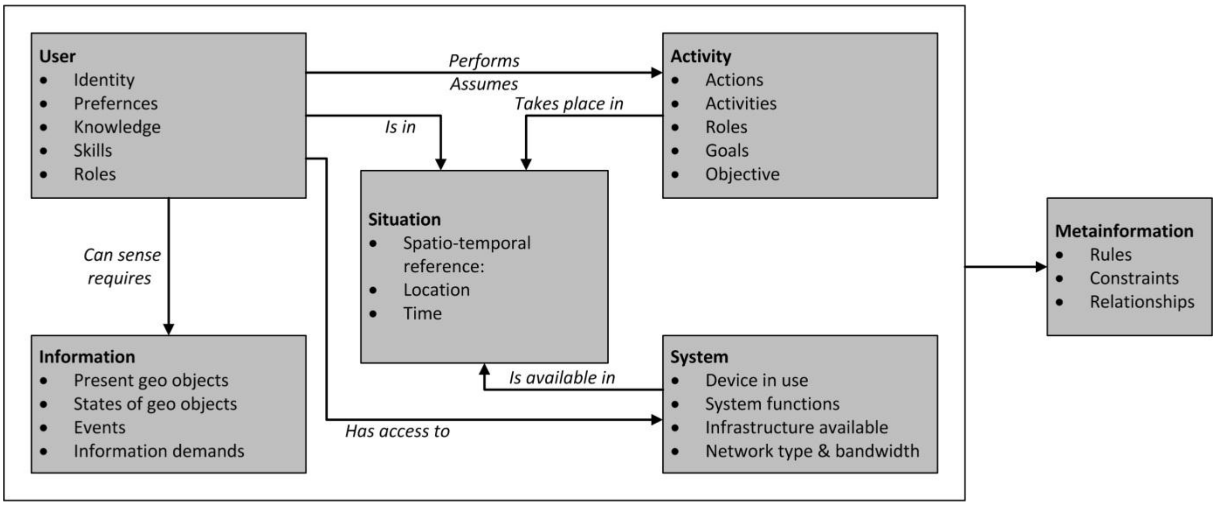


Figure AP. 2. Reichenbacher's Context Model for Mobile Cartography

As can be seen in the number of links, the user again largely defines the cartography. Adaption of the application (in particular activities, information and situation) to the user is considered to be the key factor in mobile, geographic applications. These adaptations must fit within the original intended designed application and maps. Focus on specific elements on maps can be further extended by highlighting it in colour, centring the object, emphasize it with outlines, increasing the sharpness (while making the rest blurry), enhancing the level of detail of the object and by animating the object (such as blinking, size increase and rotation) (Reichenbacher, 2005). In this project, the same interpretation and visualization can be used for texts instead.

B.3. Background information on cartographic rules

Kraak & Ormeling (2003) provided renewing interesting in the cartographic rules set by Bertin in 1967, which assembled existing conventionalities and practices of centuries of cartographic habits in a logical structure. The phrase "***How*** do I say ***what*** to ***whom*** and is it ***effective***?" is the underlying principle upon successful map designing, using cartographic rules, aside from visualization of backgrounds and the map structure itself. The ***how*** involves the use of maps, symbology, texts or charts; the ***what*** resembles the data; the ***whom*** is the target audience; and the ***effective*** implies to in what extent the three have to be combined for understanding. The *what* and the *effective* is being emphasized in this section.

Not only symbology and cartography changes with its scale (e.g. a city can be represented as a point, but also as a polygon), but the data variable itself is largely influencing how symbols should be used. Data can be nominal (classification, categorical), ordinal (when a ranking can be made), interval (where variables are distinguishable from each other by numeric values) or ratio (same as interval, except with a fixed zero point). From here, six graphical variables can be used on top of the data classification, to distinguish areas, namely by differences size, lightness/colour (value), grain/texture, hue/colour (differentiation), orientation and shape (Kraak & Ormeling, 2003, seeFigure AP. 3). Bertin related the graphical variables with the data variables to see how *effective* the representation is, as shown in Table AP. 7 and Table AP. 8.

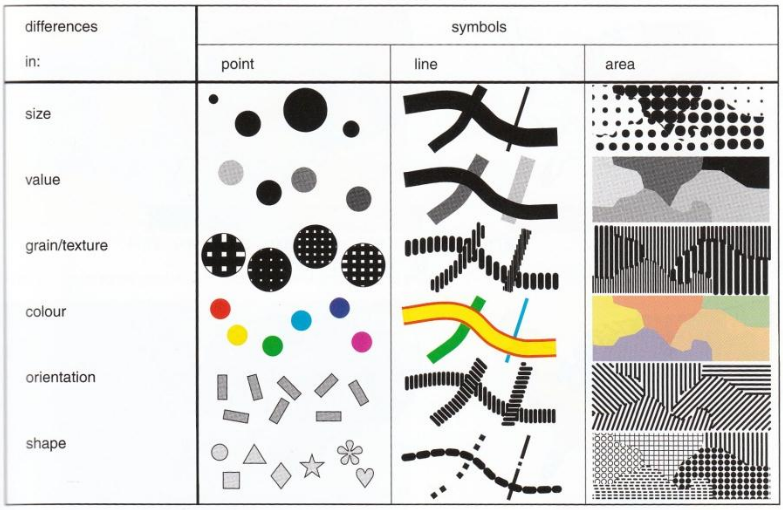


Figure AP. 3. Bertin's geographic variables

Source: Kraak & Ormeling (2003)

	Nominal	Ordinal	Interval	Ratio	Dots	Dashes	Patches
Size		X	X	X	4	4	5
Colour value		X	X		3	4	5
Grain/texture		X	X		2	4	5
Colour dif.	X				7	7	8
Orientation	X				4	2	-
Shape	X				-	-	-

Table AP. 7. Relation of graphic variables and visual isolation

Source: Kraak & Ormeling (2003)

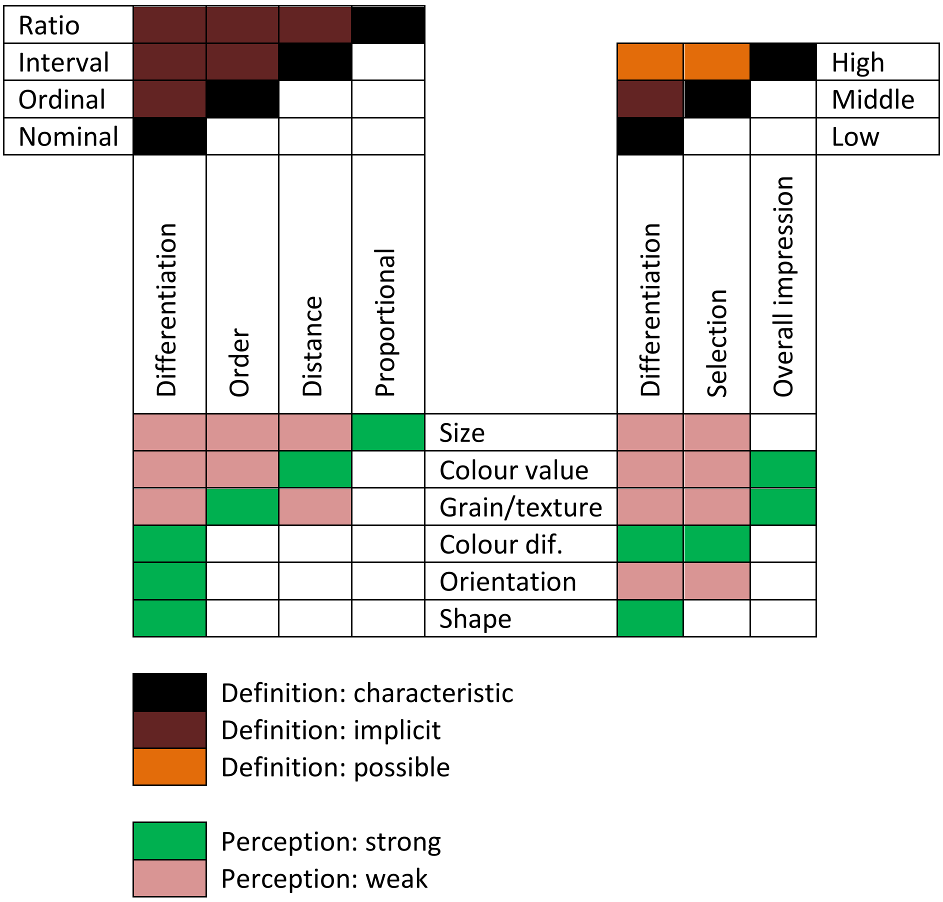


Table AP. 8. Information perception of cartographic information

Source: Kraak & Ormeling (2003)

From the above figures, Kraak & Ormeling (2003) emphasizes on the constraints of human perception in cartographic visualization. For example, there is a maximum of colours humankind can perceive to easily distinguish data values. If this maximum is exceeded, it becomes much harder to do so. In the case of colour, extra psychological and cultural associations play a fundamental role in judging a map. A red colour is usually associated with something negative, except for temperatures, where this implies higher (positive) temperatures. It can also be associated with political parties or other cultural references.

C. Results

C.1. Scan results for location estimation

The tables below provide information which location nodes have been obtained. The table headers indicate at which location the user/device was actually located.

Node	3002	3007	3011	3101	3105	3110	3203	3209	3211	2002	2005	2008
Remark	Scan ALL	Scan ALL	Scan ALL	Scan ALL	Scan ALL	Scan ALL	Scan ALL	Scan ALL	Scan ALL	Scan ALL	Scan ALL	Scan ALL
T01	2013	3107	2001	3105	2011	2001	3205	3209	3209	2004	2011	2006
T02	3003	3007	3006	2014	2011	3009	3205	3111	3211	2004	2004	2003
T03	2013	3005	2001	2012	2011	3009	3105	3108	3111	2004	2011	2004
T04	2013	3107	3112	2011	2011	3009	3209	3206	3211	2003	2004	2003
T05	2013	3107	3009	2012	2011	3011	3209	3206	3211	2004	2004	2002
T06	2014	3005	2001	3105	2011	3010	3104	3111	3111	2004	2004	2003
T07	3201	3004	2001	2014	2011	3211	2011	3108	3209	2004	2004	2002
T08	2013	3107	3009	2012	2011	3009	3205	3206	3211	2004	2004	2003
T09	2014	3005	2001	2012	2011	3011	3205	3206	3111	2004	2004	2003
T10	2013	3010	2001	2011	2011	2001	3206	3108	3212	2004	2004	2003
T11	2014	3004	3112	2014	2011	3011	3205	3111	3211	2011	2004	2003
T12	2013	3107	2001	2012	3206	3010	3205	3111	3212	2004	2004	2004
T13	3003	3004	2014	2011	2011	2001	3205	3209	3211	2004	2011	2003
T14	2014	3107	3009	2014	3206	3011	3205	3206	3212	2004	2004	2003
T15	2013	3010	3009	3105	2011	3211	3205	3111	3211	2011	2004	2003
T16	2011	3005	3011	2014	2011	3211	3205	3209	3211	2004	2004	2003
T17	2011	3004	3009	2012	3206	3009	3205	3207	3111	2004	2004	2003
T18	2013	3005	2001	3103	2011	3211	3205	3206	3209	2011	2004	2003
T19	2013	3107	2001	3202	2011	2001	3205	3209	3209	2011	2004	2007
T20	2013	3005	3010	3001	3206	3209	3205	3206	3111	2004	2004	2002

Table AP. 9. Scan results for all locations - best 10 macs in vicinity

Node	3002	3007	3011	3101	3105	3110	3203	3209	3211
Remark	SCAN FLOOR	SCAN FLOOR	SCAN FLOOR	SCAN FLOOR	SCAN FLOOR	SCAN FLOOR	SCAN FLOOR	SCAN FLOOR	SCAN FLOOR
T01	3104	3007	3011	3102	3105	3010	3207	3210	3212
T02	3103	3003	3011	3204	3105	3108	3207	3209	3211
T03	3101	3101	3011	3101	3105	3106	3204	3208	3211
T04	3003	3013	3011	3003	3105	3109	3207	3209	3201
T05	3104	3007	3005	3004	3102	3003	3207	3209	3211
T06	3104	3007	3011	3102	3101	3109	3204	3209	3204
T07	3101	3007	3003	3004	3105	3109	3206	3209	3212
T08	3101	3105	3003	3207	3102	3003	3205	3205	3211
T09	3003	3007	3003	3204	3105	3104	3207	3205	3211
T10	3003	3007	3112	3205	3105	3109	3207	3209	3206
T11	3101	3104	3108	3204	3102	3105	3204	3102	3206
T12	3104	3007	3108	3004	3105	3109	3206	3205	3204
T13	3001	3009	3112	3205	3102	3109	3204	3209	3204
T14	3103	3104	3002	3204	3101	3204	3204	3002	3204
T15	3002	3105	3002	3204	3101	3204	3204	3205	3206
T16	3104	3007	3108	3204	3105	3104	3206	3209	3201
T17	3002	3103	3011	3206	3102	3112	3206	3001	3204
T18	3003	3007	3011	3204	3101	3109	3204	3208	3206
T19	3101	3007	3011	3004	3105	3107	3204	3208	3206
T20	3002	3007	3002	3204	3101	3109	3204	3210	3203

Table AP. 10. Scan results for wing 3 locations - only routers from same floor registered

C.2. Scan results for route prediction

The table below shows the outcomes for the locations provided by the device upon walking a specific route, which has been repeated three times.

Route	1													Nodes	On	On/off	Off
1	3005	3006	3007	3008	3009	3110	3210	3211	3212					9	5	2	2
Located	3005	3004	3007	3009	3007	3109	3210	3211	3212						55,6%	22,2%	22,2%
2	3101	3102	3103	3104	3106	3107	3108	3109	3110	3111	3112			11	4	3	4
Located	3101	3105	3103	3101	3005	3203	3004	3109	3109	3112	3112				36,4%	27,3%	36,4%
3	3201	3202	3203	3204	3206	3207	3208	3209	3210	3110	3111	3112		12	6	4	2
Located	3105	3202	3203	3105	3206	3207	3208	3206	3107	3108	3112	3112			50,0%	33,3%	16,7%
4	3001	3002	3003	3103	3203	3202	3201							7	2	1	4
Located	3002	3002	3101	3103	3205	3101	-								28,6%	14,3%	57,1%
5	3212	3211	3210	3110	3009	3010	3011							7	2	3	2
Located	3212	3212	3211	3211	3009	3108	3010								28,6%	42,9%	28,6%

Route	2													Nodes	On	On/off	Off
1	3005	3006	3007	3008	3009	3110	3210	3211	3212					9	4	3	2
Located	3007	3007	3007	3008	3006	3109	3107	3211	3212						44,4%	33,3%	22,2%
2	3101	3102	3103	3104	3106	3107	3108	3109	3110	3111	3112			11	4	2	5
Located	3101	3102	3104	3105	3106	3004	3109	3109	3204	3208	3007				36,4%	18,2%	45,5%
3	3201	3202	3203	3204	3206	3207	3208	3209	3210	3110	3111	3112		12	5	5	2
Located	3202	3202	3203	3203	3206	3207	3207	3206	3108	3108	3112	3112			41,7%	41,7%	16,7%
4	3001	3002	3003	3103	3203	3202	3201							7	3	0	4
Located	3001	3002	3003	3102	3204	3204	3101								42,9%	0,0%	57,1%
5	3212	3211	3210	3110	3009	3010	3011							7	4	0	3
Located	3212	3211	3108	3108	3009	3009	3011								57,1%	0,0%	42,9%

Route	3													Nodes	On	On/off	Off
1	3005	3006	3007	3008	3009	3110	3210	3211	3212					9	3	3	3
Located	3006	3006	3006	3008	3007	3109	3210	3107	3107						33,3%	33,3%	33,3%
2	3101	3102	3103	3104	3106	3107	3108	3109	3110	3111	3112			11	4	3	4
Located	3002	3006	3102	3104	3106	3106	3006	3109	3105	3110	3112				36,4%	27,3%	36,4%
3	3201	3202	3203	3204	3206	3207	3208	3209	3210	3110	3111	3112		12	5	1	6
Located	3205	3202	3203	3203	3206	3208	3108	3108	3108	3108	3111	3112			41,7%	8,3%	50,0%
4	3001	3002	3003	3103	3203	3202	3201							7	4	0	3
Located	3001	3002	3003	3104	3204	3202	3202								57,1%	0,0%	42,9%
5	3212	3211	3210	3110	3009	3010	3011							7	2	0	5
Located	3212	3211	3107	3109	3101	3003	3002								28,6%	0,0%	71,4%

Table AP. 11. Scan results for 3 series of 5 different routes in wing 3