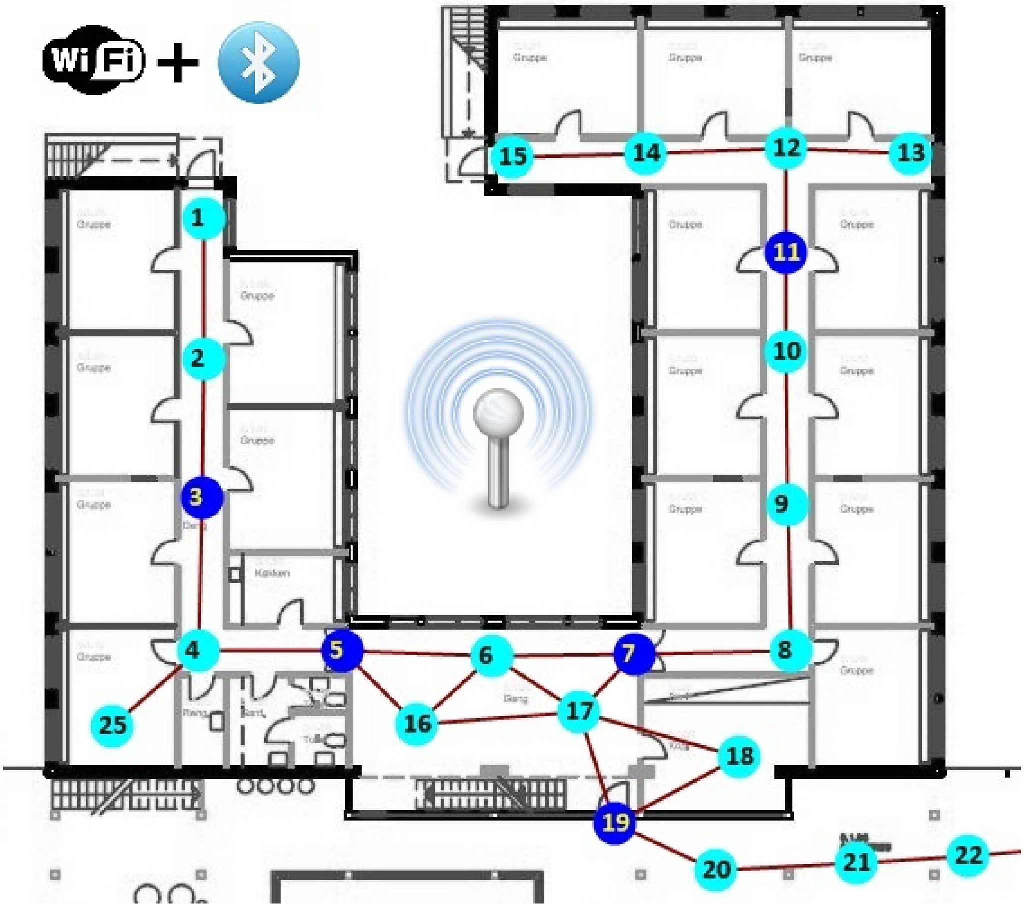
HYBRID INDOOR POSITIONING: AN EMPIRICAL STUDY IN CASSIOPEIA



Department of Computer Science
Aalborg University
June, 2010

Department of Computer Science

Aalborg University

Selma Lagerlöfs Vej 300
9220 Aalborg Ø
+45 9940 9940
+45 9940 9798

PROJECT TITLE
Hybrid Indoor Positioning: an Empirical Study in Cassiopeia

PROJECT PERIOD
01.02.2010 - 18.06.2010
DE4

PROJECT GROUP
d625a

GROUP MEMEBERS
Artur Baniukevic
Dovydas Sabonis

SUPERVISOR
Hua Lu

NUMBER OF COPIES: 2

PAGES IN REPORT: 89

PAGES IN APPENDIX: 4

PAGES IN TOTAL: 109

ABSTRACT

The aim of this project is to investigate possibilities of combining distinct wireless technology standards into one indoor positioning (IP) framework and find ways of making it more adaptable to inconsistent environments and different user device specifications.

Throughout the project we explore possibilities of how to combine advantages of Wi-Fi and Bluetooth based IP methods into one integrated framework.

As a result, implementation of such framework is designed, developed and deployed. A number of additional techniques are then utilized which enables the IP to reach higher level of accuracy and precision.

Preface

This report was written during the 4th Data Engineering semester at Aalborg University Computer Science department, spring 2010.

We would like to thank *Hua Lu* for being our great supervisor and giving us useful advices. Individual acknowledgment we give for *Rene Hansen*, who introduced Wi-Fi based positioning details. Many thanks to *Jesper Brix Rosenkilde* who provided information about WLAN network specifications in our department.

Content of attached CD:

- PDF version of this report
- Source code including SQL scripts and dynamic-link libraries

Signatures:

Artur Baniukevic	Dovydas Sabonis

Reading Guide

This report is organized into the following chapters:

Resume: provides a brief introduction to the research area that we were working on. Briefly defines what goals we were trying to achieve and what problems we have faced. Finally a list of our contribution to this research area is presented.

Positioning Techniques and Technologies: shortly covers available technologies and indoor positioning techniques. Summary of advantages and disadvantages of different methods is presented.

Design and Implementation: presents the design of our indoor positioning system together with implementation details.

Experiments: is the biggest and most important part of this work. It starts by presenting the equipment that was used as well as environment that the experiments were conducted in. Later it describes a number of techniques that were developed in order to solve the problems defined in the beginning of this report. Description of each technique together with experiment results and brief conclusions are presented as well.

Conclusions: gives a summary of the results that were achieved together with advantages and disadvantages of each method. List of suggestions for possible future work are also presented.

Appendix A Figures: less important, supplementary figures can be found in this chapter.

Appendix B Terms and Abbreviations: contains a list of essential terms and commonly used abbreviations throughout the report. Upon encountering unfamiliar abbreviation the reader should refer to this section.

We conclude this report by presenting the list of references.

Contents

List of Figures

List of Tables

Chapter 1

Resume

The recent decade has witnessed an increased interest in various location-based services. While outdoor positioning systems have already matured – new location-based services that work indoor or both - indoor and outdoor, are one of the more popular research areas of many companies and scientists. A few properties of indoor space makes it difficult to use with the existing positioning techniques that are being used outdoors. First of all - indoor space is composed of many entities (walls, rooms, doors, etc.) that restrict the movement of objects. The most commonly used positioning techniques such as GPS cannot be applied in an indoor environment because the signal is usually blocked from reaching client's device. Furthermore, as walls tend to block and reflect incoming signals – it makes it very difficult to report position with fairly high level of accuracy and techniques such as those used in GPS are incompatible.

A lot of research has been carried out in order to find new accurate positioning techniques that would enable users to identify their position in an indoor space utilizing the existing infrastructures such as Wi-Fi, Bluetooth, Infrared or RFID readers. Therefore, instead of discovering new or improving indoor position estimation methods like triangulation or position fingerprinting 2.1.3, this project's main focus is directed towards increasing indoor positioning system's precision and performance by finding new ways of combining different techniques and infrastructures into one integrated system. Ideally – indoor positioning system should be self-maintainable, only requiring user input when absolutely necessary. Furthermore the system should try to combine advantages from different available infrastructures and be able to adapt to different inconsistent environments and distinct client device configurations. The system should also aim to enforce physical constrains of a floor map (walls and other obstacles) in order to minimize possible position miscalculations. Finally - even after meeting all the previously described requirements, the positioning system should still be affordable for consumers with lower budget requiring minimal additional equipment to be purchased.

Those are the main requirements that this project is focusing on.

Project goals

Throughout the whole duration of the project we were focusing on achieving the following goals:

- Utilize available infrastructures and integrate indoor positioning technologies into one system
- Design indoor positioning model adaptable to distinct client device configurations and current environment.
- Incorporate self-maintainability (updated fingerprints are collected automatically as described in 4.7.6.2)
- Utilize past trajectories of users (history)

Problem statement

We have encountered the following problems when trying to achieve the goals mentioned above:

- Clients with distinct wireless interface specifications reporting different signal strength indications
- Dealing with changing/inconsistent environment where signal strengths vary throughout the course of the day
- Jumping through walls issue [9]
- Optimal Bluetooth station deployment/placement

In the later chapters of this report we propose solutions to these problems and present results indicating how efficiently the problems were solved.

Contribution

The following contribution and accomplishments were achieved throughout the duration of the project:

- Advantages of two distinct wireless infrastructures were integrated into one indoor positioning system as described in section 4.7.6.
- Proposed an indoor positioning model which attempts to adapt to different client device specifications as described in section 4.7.3.
- Technique described in 4.7.1, similar to those described in works [9], [10], that utilizes a graph structure in order to enforce physical constraints of an indoor space was implemented and integrated into the system. Our approach slightly differs from the mentioned works as we introduce a reachability parameter explained in section 4.7.1.

- Designed flexible indoor positioning model presented in 4.7.2 which tries to adapt to inconsistent environments where signal strengths tends to vary during the course of the day.
- Proposed a method 4.7.4 that utilizes past trajectories of users during the position estimation procedure
- Indoor direction prediction framework described in 4.7.5 was implemented
- A complete indoor positioning system with graphical user interface encapsulating all the above mentioned features was designed, implemented and deployed

Chapter 2

Positioning Techniques and Technologies

This chapter elaborates on different wireless technologies and existing positioning techniques which form the core of the indoor positioning systems. Advantages and disadvantages of these techniques and technologies are revealed in this chapter.

2.1 Position estimation techniques

In this section general positioning techniques are briefly presented based on [24], [7], [18], [30], [9], [16]. The choice of particular methods is explained and emphasised.

There are mainly three position estimation techniques widely used in positioning systems: triangulation, proximity and scene analysis.

2.1.1 Triangulation

Triangulation is geometric approach of position calculation. It uses properties of triangles to compute object locations. This technique can be divided into lateration and angulation methods. In lateration - time-based distance measurements are utilized, while in angulation angle-based distances are employed [24]. Distance refers to distance between receiver and transmitter. In lateration there should be at least three transmitters in order to be able to estimate correct position [18].

In indoor space there is almost no line-of-sight (LOS) path between transmitter and receiver. Radio signals reflect off different surfaces and obstacles creating multipath issues. These issues make triangulation based position estimation methods which uses time- or angle-based measurements cause inaccuracy [17]. Triangulation method is used in outdoor space more successfully. Triangulation is not very popular in indoor environment as it requires knowing locations of transmitters, in WLAN positioning - location of access points, which usually are not available. Moreover in case of angulation special receivers are required which are capable to get angle data from transmitters [30].

2.1.2 Proximity

Proximity method presents position estimation based on closeness of mobile user actual location to transmitter location ([18], [24]). In this technique every transmitter is associated with particular position. When mobile user is in range of such transmitter then user's position is associated with position of transmitter. If more than one transmitter is detected, position of transmitter with strongest signal is selected. This proximity method is efficiently used with low range signal transmitters like RFID or Infrared. Disadvantage of this method is that it requires lots of transmitters to be in the area in order to make positioning very accurate.

Proximity method was used with low range Bluetooth receivers in our project due to its simplicity and no computations required. When Bluetooth station is detected position of this station is chosen.

2.1.3 Scene Analysis

Scene analysis methods describe positioning which is based on prior collected signal data and is more often called fingerprinting. Comparing to triangulation and proximity this technique does not require knowing locations of transmitters. Fingerprinting technique that we have chosen for WLAN is based on received signal strengths. These signal strengths obtained from all transmitters available in certain range are utilized in position calculation. In this technique two phases exist: offline and online, also called training and positioning. Offline phase always comes before online. In training stage fingerprints (collection of signal strengths) are recorded in user-defined positions in the environment. As a result database of fingerprints known as radio map is built. During the online phase mobile user reports vector of received signal strengths which later is compared with fingerprints in radio map and in consequence position associated with the best matching fingerprint is returned to the user. In order to find the best matching fingerprint two general models exist: deterministic and probabilistic [13]. In deterministic model actual values of signal strengths are used comparing to signal strength probability distribution in every reference point used by probabilistic method. Probabilistic model was not used in this project due to its computation complexity and lower accuracy compared to deterministic [17]. As a deterministic model we employed Nearest Neighbor in Signal Space method which utilizes Euclidean distance formula to compute distances between fingerprints in database and incoming signal vector. The closest fingerprint is defined where smallest distance is detected.

It is worth mentioning that fingerprinting technology is not perfect. The main drawback of fingerprinting method is that it is highly dependent on transmitters' infrastructure. If locations of transmitters are changed, radio map needs to be updated. And of course offline phase is very time consuming process if indoor area is reasonably wide.

2.2 Different wireless technologies

This section briefly describes today's most commonly used wireless technologies that can be utilized by indoor positioning system.

2.2.1 Wi-Fi (IEEE 802.11)

Wi-Fi is probably the most commonly used infrastructure in indoor positioning systems. It is mainly because of the fact that it has become a standard method of communication and is now available in most of today's institutions and private apartments. No extra equipment is needed in order to deploy an indoor positioning system in an environment which already has a decent Wi-Fi coverage. The solution in that case can be based on software only. Another advantage of this infrastructure is that it scales easily because of its rather high radio coverage – usually up to 150 meters in an indoor space and 300 meters outdoors. As a possible disadvantage we could mention that this infrastructure requires a relatively large amount of power/energy. According to [2], compared with Bluetooth – Wi-Fi consumes approximately 5 times more energy. This might be a potential problem in some mobile devices where energy consumption should be kept to minimum.

There is a number of previous works [1], [29], [9] written on building an indoor positioning system based on Wi-Fi infrastructure.

2.2.2 Bluetooth

Bluetooth is another very popular wireless technology standard. It was designed mainly for voice and data applications and is integrated into most of today's portable electronic devices such as cell phones PDAs and other. Specifications of Bluetooth are very similar to those of Wi-Fi. The advantage of this wireless technology over Wi-Fi is that it is more targeted towards smaller devices and hence uses only a small portion of energy compared with Wi-Fi. This way a user can use indoor positioning system installed on his mobile phone for a much longer period of time. There are a few different classes of Bluetooth radios [2] as shown later in table 4.6. This specification defines the radio coverage of a Bluetooth device. It can vary from 1 to approximately 100 meters depending on the class. Another advantage compared with Wi-Fi is that Bluetooth equipment is usually up to 3 times cheaper. However the higher range of Wi-Fi can sometimes compensate that. A more detailed explanation about indoor positioning technique that was designed on top of Bluetooth technology can be found in [26].

2.2.3 Active Radio Frequency Identification (RFID)

The RFID can be defined as a lower range wireless infrastructure. It is usually composed of a reader and a number of tags that can be scanned and identified while being in the range of the reader. The typical range of this infrastructure is usually between few centimeters up to a few meters. There are some devices capable of getting the signal from up to 100 meters but those are rather rare and expensive. The advantage of this technology is that it is able to quickly identify different objects carrying different tags based on the information received from the tag. The fact that RFID is not as popular as the other two wireless technology standards described above means that usually some additional equipment needs to be purchased before deploying an indoor positioning system based on RFID.

There are a number of previous works [10], [11] written about RFID based indoor positioning systems.

2.2.4 Infrared (IR)

IrDA [8] is another short range wireless communication infrastructure which exchanges data over infrared light. The typical range of this infrastructure is approximately one meter. The biggest limitation compared with wireless communication technologies described above is that infrared signal does not penetrate solid materials. It means that the sender and receiver should have a line of sight for the communication to be successful. Despite this limitation there are a some indoor positioning projects [3] that successfully utilize this technology.

2.2.5 Summary of different technologies

Indoor positioning systems based on short range wireless communication standards such as RFID, IrDA and part of Bluetooth determine position of an object based only on object's presence in a particular area which is within range of the scanner/receiver. Whenever an object gets in range of such device we can determine its position with a rather high precision. By knowing the position of the actual device we can be sure that if the object was detected by the device – it will be within one meter (depending on range of the device) from that position and no further location calculations are necessary.

Using this approach we can determine precise position of an object. However if we were about to cover a larger indoor area - it would require a great number of such devices to be purchased and deployed which is both not practical and expensive. On the contrary – Wi-Fi infrastructure can cover a relatively large area but at the same time making it difficult to calculate the exact position of an object. Additional techniques such as position fingerprinting described in section 2.1 need to be employed in order to estimate approximate position of the object. This in turn does not guarantee that position will be estimated with high precision. Typical precision of indoor positioning systems based solely on Wi-Fi is approximately from 1 to 6 meters in a stable environment with no wireless signal strength deviations. In reality – precision may sometimes drop depending on many factors that have some influence on wireless signal's strength variation. Because of the nature of wireless signals and complexity of indoor environment, indoor positioning system can sometimes produce very inaccurate results. Indoor space is usually composed of a variety of objects and obstacles (walls, floors, doors, etc.) which tend to block or reflect wireless radio signals. Furthermore, some wireless access points are designed in such way that they increase wireless signal power level according to the current load/bandwidth. There are also many other physical factors, such as humidity, that may one way or another affect the quality of received wireless signal. Such instability has a noticeable negative effect on the precision of indoor positioning system based on high range wireless infrastructures such as Wi-Fi because position determination is usually estimated by assessing received wireless signal strength values as described in section 3.5. On the other hand - short range wireless communication infrastructures are immune to such factors but are usually more expensive to cover the larger indoor areas.

As pointed out in the paragraph above – the choice of using Wi-Fi or a low range wireless communication standard for indoor positioning has its advantages and disadvantages. Finding a novel way to combine two distinct infrastructures into one unified system would enable us to have a more robust indoor positioning framework that could combine the advantages from both approaches. As mentioned in the previous sections – this is one of the main ideas behind this project. In the latter sections we propose a way of how to integrate Wi-Fi together with low range Bluetooth infrastructure into one indoor positioning framework.

Chapter 3

Design and Implementation

3.1 Designing relationship between entities

In this stage of our work we have decided on what entities our system will support and what relationships will be preserved between them. In this case we have designed entity relationship diagram which later was extended and converted to relational database schema 3.2. Designing such diagram gives better overview of the system and data which needs to be handled.

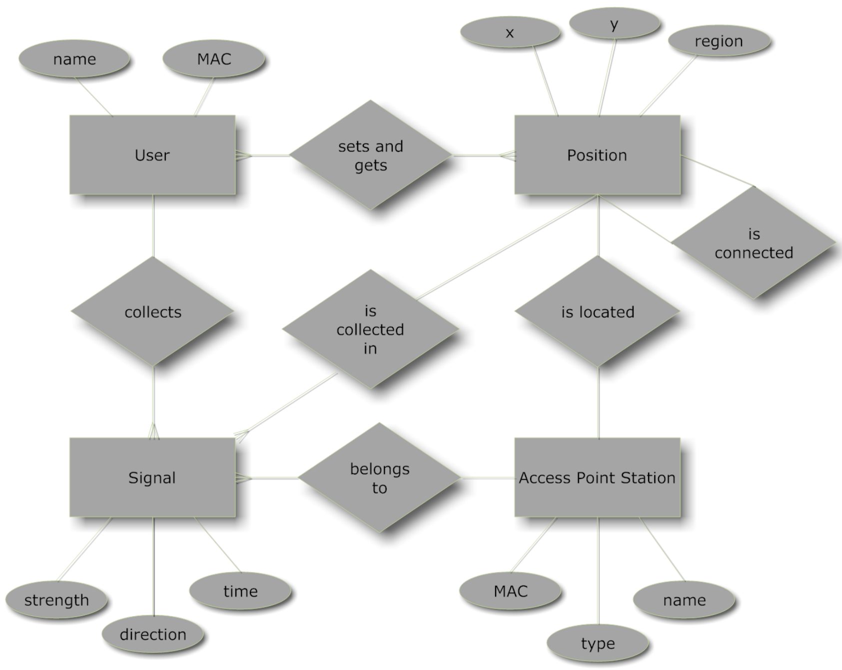


Figure 3.1: Entity relationship diagram

As it is shown in the figure 3.1 four main entities were considered in our positioning system: user, position, signal and access point. User entity refers to different mobile devices (distinguished by MAC address and name) which are being tracked or which were used as signal collection contributors. Collection contributors can be also a part of a tracking process. Requirement for these mobile devices is that WLAN network card and Bluetooth interface should be available.

Every signal is collected by user in certain position and specific time. Signal is specified by strength expressed in dBm. Direction attribute is designed for direction in which signal was collected. Signal is received from the certain access point station. There are two types of access point stations we use: Bluetooth and Wi-Fi. In case of Bluetooth access point station we do not care about the signal specifications received from this station. Position on the other hand can be a location of fingerprint or location of access point. In the system we consider the fact that different users can collect signals in the same position. One position can be associated with lots of signals collected by different user. Every position belongs to the certain region, which denotes particular segment of the building. Regions can also help to distinguish positions on different floors. As our aim is to employ graph structure in our system, position must be connected with other position as explained in 4.7.1.

3.2 Data model

In this section database model is described which is converted from entity relationship diagram. During the system implementation process model was extended and additional tables were added to the model in order to facilitate data manipulation, querying and modification. In general, model can be divided into connected and isolated tables. Connected tables make fundamental part of data model. They are linked between themselves by foreign keys. While isolated tables have more facilitation function: they are employed for data backup, temporary results and storing bulk data. Isolated tables and less relevant or important table fields will not be described deeply.

In database schema in figure 3.2 connected tables are represented. Entities from ER diagram 3.1 are converted to appropriate tables in the schema: *users*, *signals*, *positions* and *aps*. For convenience purpose every table contains *id* which is primary key. Every foreign key can be identified by notation pattern: "referencing table name abbreviation" + "_id". In database schema we use foreign keys with "ON DELETE CASCADE" option. For example: if particular position is deleted record from positions table, records in the tables referencing positions table are no more valid and therefore also removed.

As a result of M:N relationship between *users* and *positions* table *userpos* is created. This allows storing user and position *id* without redundancy in the referenced tables. In this table those users are saved which have collected signals in specific positions in offline phase. Combination of *u_ id* and *pos_ id* is always unique in this table.

In *aps* table two types of access points are stored: Bluetooth and Wi-Fi. Bit type *iswifi* field stands for separating these types. Every access point has a unique MAC address field *mac*. Field *name* stores SSID (network name for WLAN access points) or name of the Bluetooth device. *pos_ id* is usually associated with position of Bluetooth devices. Wi-Fi access points locations are not utilized in our system. Keeping further in the *maps* table different floor maps are associated: *map* identifies location of the map file in the system, *floornr* – number of the floor. Only chosen Bluetooth devices which are checkpoints between the floors are associated with the map, in other cases *map_ id* in *aps* table has null value.

In one of the experiment cases we use graph model where positions are connected between themselves. Table *connections* correspond to an edge entity in the graph model. Only two fields are required to represent it: starting position *pos1* and ending position *pos2*. Edge is considered as the shortest path between two positions. In this table only possible edges are stored respecting topology of the building. Moreover as we do not use directional graph two edges (x,y) and (y,x) are considered as the same and are treated as a duplicate in this table.

In order to be able to query user history data, *poshistory* is added to the database model. As this table is more like bulk table, in other words huge data storage is involved, to make querying and inserting faster we do not connect it with *userpos* table but referencing *users* and *positions* tables directly is done instead. Otherwise using *userpos* table, before inserting data it will be necessary to check if combination of particular *u_ id* and *pos_ id* exists, and perform more join operations, which requires more time and recourses. On the other hand *userpos* offline phase data is not mixed with history in order to keep it less confusing and complicated.

Based on data from *poshistory* table most of the experiments evaluations are made. Fields *pos_ id*, *true_ pos_ id*, *changedbypred_ pos_ id* are used as parameters in these experiments evaluations: *pos_ id* - identifies position which was estimated by algorithm and shown on the map, *true_ pos_ id* – identifies position which is real position of the moving object defined by user or program, *changedbypred_ pos_ id* – identifies position which was modified based on historical data. Field *error_ dist* presents error Euclidean distance between estimated position and true position. *Direction* refers to online direction of the moving user, while *predict_ direction* – direction predicted by formula which will be described in experimental part. Every historical record is time stamped in *htime* field up to milliseconds, has a number of access point seen at that current moment (*apcount*), and average signal strengths (*avgstrength*) from all access points moving user could see at the particular time in the past. *Iswifi* refers to the type of visited position estimation either based on Bluetooth scanning or WLAN scanning results.

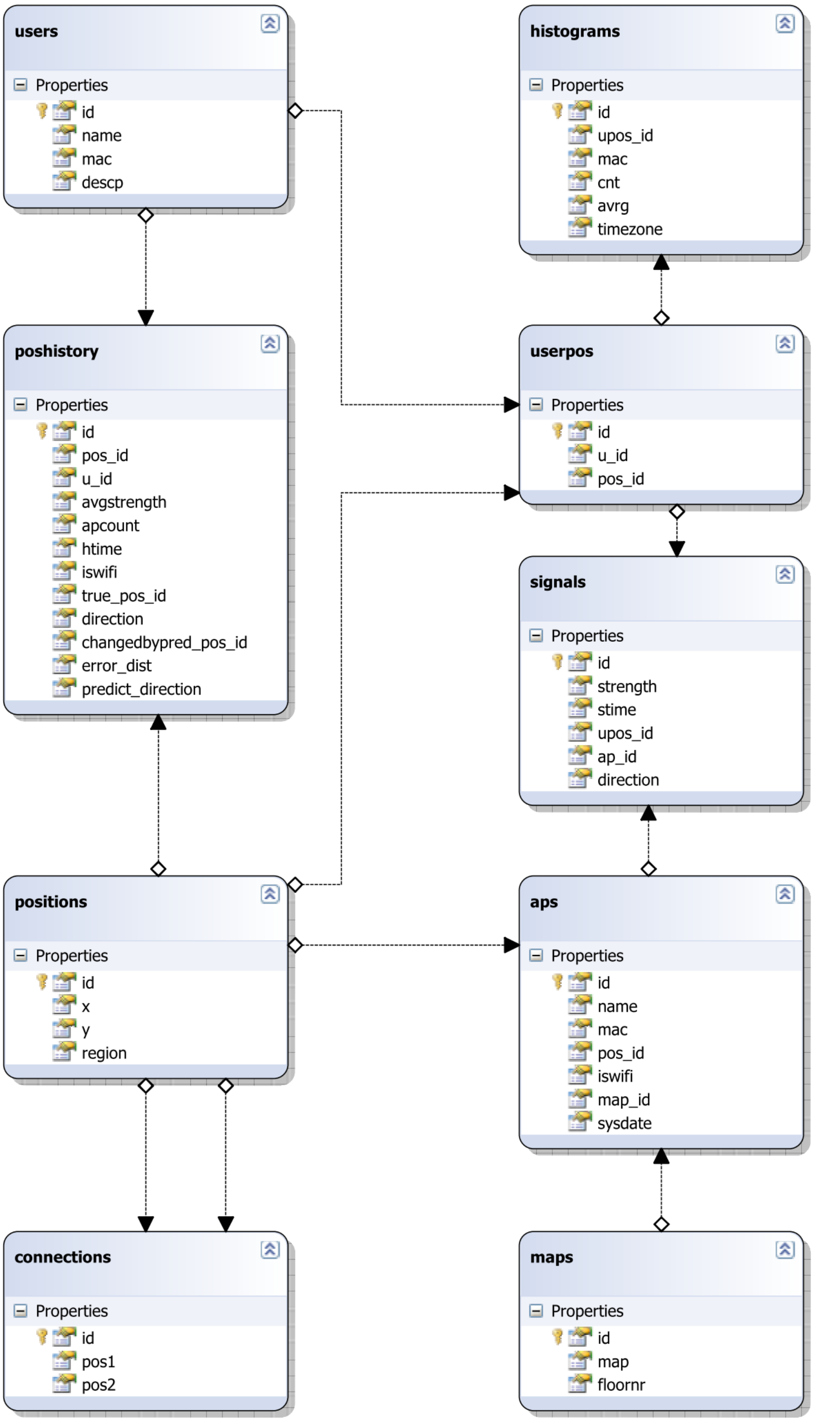


Figure 3.2: Database schema

Finally to achieve higher performance we added *histograms* table to the model. We define histograms as average signals strengths (*avrg*) grouped by MAC address of access points. *Timezone* field defines time interval signal belongs to. It is used in experiments with the fingerprint splitting by time. To avoid repetitive aggregate queries used in position estimation method 1 (line 8) we use histogram approach instead. Performance increases rapidly and result of comparison can be seen in figure 3.3. As we can notice, while using histograms position is computed almost 6 times faster than grouping signals every time separately. Additionally when more signals collected in one position and more reference points available in the system, histograms value increases, as it saves system and especially database, resourses including time 3.3. Histograms table is populated whenever new signals are collected in offline phase.

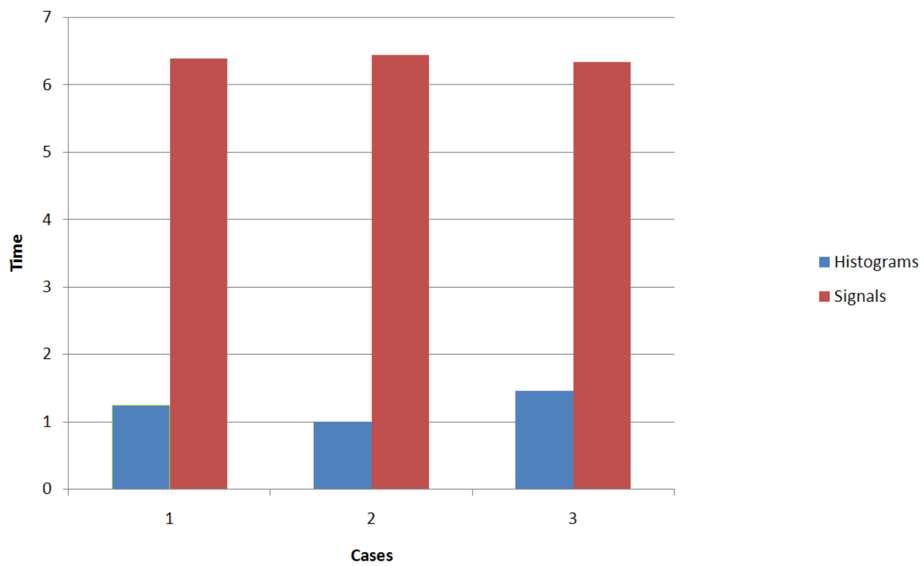


Figure 3.3: Performance using histograms

Isolated tables in database are *test_ signals*, *train_ signals*, *online_ signals*, *statisticsFirstZone*, *statisticsSecondZone* and *evaluation*. Three first tables have the same structure. Two of them are used in simulation experiments. Data from both tables is not overlapping among themselves. *online_ signals* table is used for collecting data from online phase for testing purpose. *statisticsFirstZone* and *statisticsSecondZone* are temporary tables employed while dividing signals data into train and test, when time intervals are set.

3.3 System architecture

The system is designed as depicted in figure 3.4.

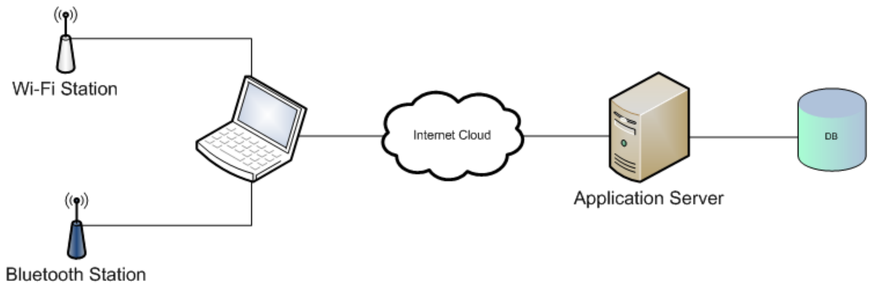


Figure 3.4: System architecture

User's portable device is used to continuously scan and upload available Wi-Fi access point and Bluetooth station signal strengths to the server. Communication between the client and server is implemented in the form of a web service. Clients send the most recent scan results to the server and get an estimated position as a result. This way any device with internet connectivity can successfully use the system despite of its operating system and other factors.

Server stores and retrieves location fingerprints in the database. The application and database server in our case is on the same physical machine.

3.4 Technologies used

Our system implementation is based on the following technologies:

Role	**Product**
Programming language	C#
DBMS	Microsoft SQL server 2005
Wireless network APIs	Native Wifi [22] and NDIS [25]
Bluetooth API	32feet.NET [19]

Table 3.1: Technologies used throughout the project

The choice of using the following products was dictated by the fact that the application server in our department was running Microsoft Internet Information Services (IIS) version 6.0 and had SQL server 2005 installed as a database management system.

Because we were dealing with machines with different operating systems we were using two different wireless communication APIs. Microsoft *Native Wifi* API was used on the machines with Windows 7 while Network Driver Interface Specification (NDIS) wrapper library was used on the machines with Windows XP operating system. Both of those APIs were utilized in an open source wireless network scanning utility by *metageek* called inSSIDer [21] licensed under *"Apache License, Version 2.0"* free software license. We have compiled part of this code into a dynamic link library and used it while implementing our system.

The API that was used for Bluetooth communication part - 32feet.NET - is developed by *In The Hand Ltd.* We were using in on top of Microsoft Bluetooth protocol stack.

3.5 Position estimation method

This section presents position estimation method which is used in our system.

In order to find the closest match in the system's radio map, nearest neighbor in signal space (NNSS) technique is used which is commonly used in the different positioning systems [1] due to its simplicity and reasonable accuracy. The point of NNSS is to compute distance between signal strengths associated with particular position and online scan's signal strengths and return the minimal distance. To compute the distance between fingerprints p and q we use Euclidean distance formula 3.1:

$$distEucl(p, q) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2,\, p = (p_1, p_2...p_n)\, and\, q = (q_1, q_2...q_n)} \quad (3.1)$$

In most cases distance between fingerprints correlates with physical distance between fingerprint positions, lower distance between fingerprints in most cases means lower distance between positions of fingerprints. However often some fingerprints may be very similar with fingerprints which are located in different places. That is why we apply additional techniques to control such situations.

Our position estimation method is based on Euclidean distance calculation. Pseudo code for this method is presented in the following method 1. In this method only main and simplified operations are included. In the simplest version of this method in order to receive best position all radio map has to be checked (line 4-21). In modified version of the method we also utilize possible reachable graph nodes 4.7.1 and regions 4.7.6 to limit search space. Going further, for every search position signals are selected (line 5). To make it clear, we map a signal to one record in *signals* table as shown in figure 3.2. Every signal record is associated with MAC address and signal strength obtained from AP. In algorithm we use two functions to present connection between these entities: MAC(signal) = {signal} → {MAC address} and SS(signal) = {signal} → {signal strength}. Due to that fact that in the offline phase we perform more than one scan, there are lots of MAC duplicates associated with one position and that's why we use average value of SS belonging to particular MAC (line 8).

Function `estimateLocation(onlineSignals)`

```
   Output: bestPosition
1  dist ← 0;
2  bestPosition ← 0;
3  minDist ← max integer value;
4  for ∀ position ∈ radiomap do
5  |  dbSignals ← set of signals in DB associated with position;
6  |  for ∀ onlineSignal ∈ onlineSignals do
7  |  |  if MAC(onlineSignal) ∈ {MAC(sig) | sig ∈ dbSignals} then
8  |  |  |  avgss ← average signal strength of dbSignals associated
   |  |  |  with MAC(onlineSignal);
9  |  |  |  dist ← dist + (SS(onlineSignal) - avgss)^2;
10 |  |  end
11 |  |  else
12 |  |  |  dist ← dist + PENALTY;
13 |  |  end
14 |  end
15 |  dist ← √dist;
16 |  if dist < minDist then
17 |  |  minDist ← dist;
18 |  |  bestPosition ← postion;
19 |  end
20 end
21 return bestPosition;
```

In 6-15 lines of the method 1 Euclidean distance is applied with small modification. We cycle through one scan's signal set collected in online phase and check if there is such a signal record in database with same MAC. In case when such record is not found we apply penalty. Penalty is used to increase the distance and decrease proximity between SS vectors. Experimental results showed that using penalty, estimation results are improved. More details about penalty determination and its impact can be found in the section 4.6.1. Finally the method returns best position where computed distance in signal space is lowest (line 16-19).

3.6 Application specification

This section briefly describes the process of client – server communication and provides some specific details of how application is implemented.

3.6.1 Client - server communication

Using the system can be divided into three phases as illustrated by figure 3.5.

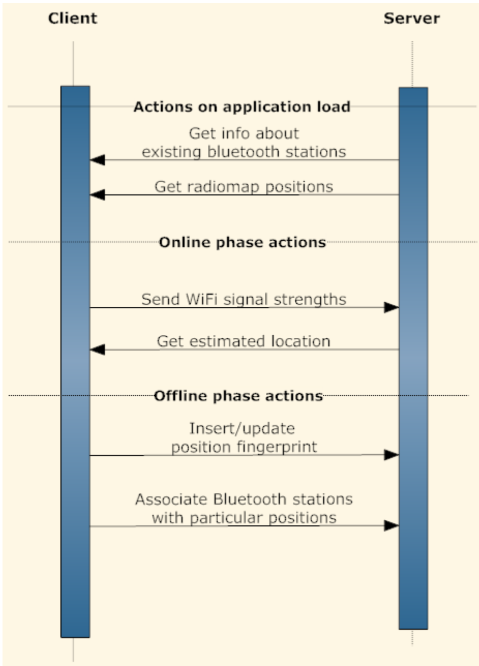


Figure 3.5: Client - server communication

First, upon starting the system the client needs to download the list of deployed Bluetooth devices and coordinates of reference points that are already in the database. We decided to download Bluetooth device list to the client because even in situations where there is no or low Wi-Fi signal coverage – the client can still estimate his position based on Bluetooth and without having a stable connection to the server.

During an online phase we have a two directional communication. The client regularly sends its Wi-Fi signal strength indications to the server where the most probable position is estimated and delivered back to the client.

There are a few actions that can be performed during the offline phase. The clients might insert a new reference point and upload the position coordinates together with a list of recorded signal strengths to the server. It is also possible to update the fingerprint of already available reference points – the newly recorded signal strength values are then stored in the database together with the initial fingerprint. This way users can maintain updated radio map throughout the longer periods of time.

During the offline phase it is also possible to deploy additional Bluetooth devices. It is done by the client by first creating a reference point and later telling the server to associate it with a MAC address of the specific Bluetooth device.

3.6.2 Multithreading

The client side of the system consists of several threads running in parallel. There are three main threads designed for specific tasks. One of the more important threads that form the core of our system deals with continuously scanning and recording available Wi-Fi signal strengths. Similarly, another thread continuously scans the area for available Bluetooth devices. Wi-Fi scanning is usually performed once each second while Bluetooth scan is performed once every 3 seconds. This highly depends on the drivers of the interface. The main thread is responsible of uploading the scanned result set to the server and retrieving the estimated position. It also calls the redraw method which updates user's current position on the interactive map. The diagram of such parallel operations is depicted in figure A.3 in the appendix section.

3.7 User interface

In order to control our system more conveniently we have designed graphical user interface as shown in figure 3.6. This makes application very interactive and comfortable to use. Most of the parameters available in the system can be set in the interface and handled by the core of the application.

In this paragraph we will briefly describe the interface presented in figure 3.6. There are three main zones on the interface: online signal list (1), parameter setup, execution and status (2, 3) and map panel. All of them except 1 are input zones, where some information can be passed through interface to the server. In the 2 and 3 zone we are defining values of parameters. By checking one of the checkboxes with specific value beside, we invoke one of the cases presented in experimental part. Offline phase and online phases are coordinated by pressing appropriate button: Insert or Track.

On the map panel user can easily set, select and delete positions. After position is stored in the database it can be connected with other positions. Bluetooth station and scanning result sets can be associated also only with stored positions. Colors of the points define purpose of the position: green position means true or estimated position, blue means position where Bluetooth station is located and cyan position is just stored position in database. When tracking is activated green point starts to show the estimated position. In experimental mode true position (green point) is selected by user and later on results evaluated after pressing Evaluate button. Changing the floor number will change the map. Positions from the first floor would not be redrawn on the second floor.

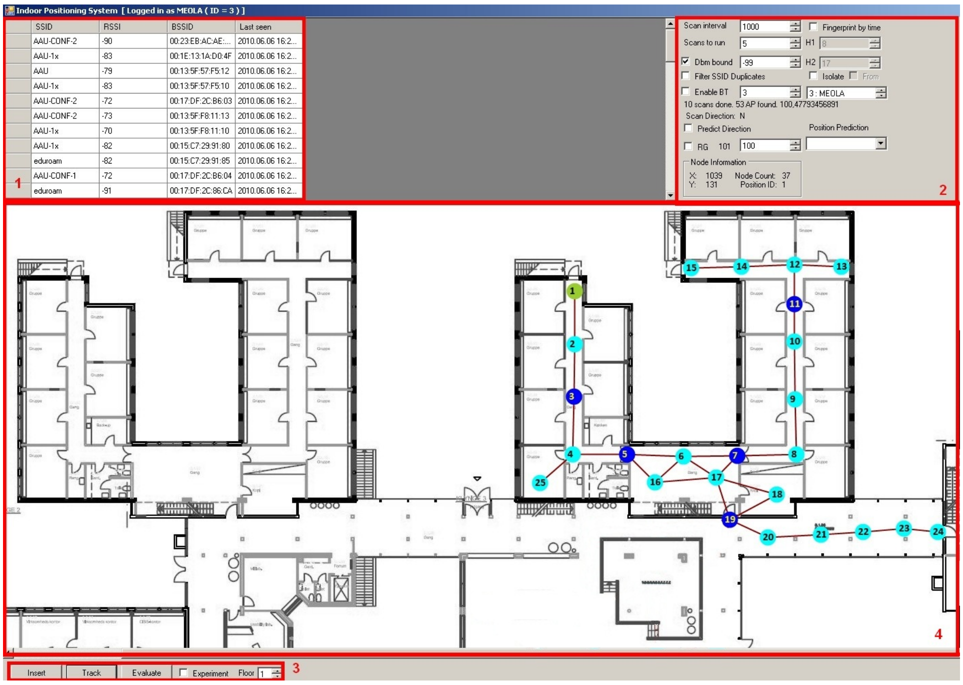


Figure 3.6: Graphical user interface

We have added Google Maps extensions to our application. Mainly our program supports raster maps which are loaded from the graphics file. However this extension can be useful in case when raster map is not available or to make tracking information easier accessible on the Web. Instead of raster map we load web mapping service Google Maps. As we have few approximate mappings between marginal pixels and coordinates (longitude, latitude), pixel measurement of estimated positions are converted to longitude and latitude coordinates.

Chapter 4

Experiments

This chapter describes how experiments were carried out and what results were achieved. Before presenting end results of each experiment that was conducted - we first define the circumstances that could have had an impact on the accuracy and performance of the system. In section 4.1 we describe what equipment was used throughout the whole experiment phase. The following section 4.2 defines the environment that we were doing the experiments in. Sections 4.5 and 4.3 explain how fingerprints were collected and what measures were used to evaluate the outcome of each experiment. Sections 4.4 and 4.6 then explains different types of experiments that were carried out and explains a number of parameter values that the system was using while running and evaluating the experiments. Finally, section 4.7 is divided into a number of subsections that present description and illustrate the results of each experiment.

4.1 Equipment

In this section we describe what equipment was used throughout the whole project.

4.1.1 Wi-Fi

Table 4.1 gives a summary of the laptops that were involved in either collecting the location fingerprints or testing the indoor positioning system.

Laptop	Operating System	Network Interface Controller	Connectivity
Asus EeePC 901	Windows XP	Realtek RTL8187	USB 2.0
HP Compaq nc6000	Windows 7	HP WLAN W400	Mini PCI-e
Compaq nw8440	Windows XP	Intel(R) PRO/Wireless 3945ABG	Mini PCI-e
Acer Aspire 9512	Windows 7	Intel(R) PRO/Wireless 3945ABG	Mini PCI-e

Table 4.1: Devices that were used in the project

The main factor that determines how accurately system can predict location of the specific user is closely related with what Network Interface Controller (NIC) is used to collect the signal strengths of all the available access points. There are a large number of different wireless NICs available on the market nowadays. Each has its strengths and weaknesses – hence they are all unique and you cannot expect that controllers that were made by different manufacturers will have identical properties and functionalities. We derive this speculation from our own experience with different types of portable devices.

In case when you are using indoor positioning system with a NIC which was not used during the location fingerprint collecting phase – system accuracy usually drops. This is due to the fact that different devices tend to record different signal strength indications and have different antennas which determines how large the scanning range is.

As shown in table 4.1- we had a chance to experiment with three different NICs. Asus EeePC 901 was the device that we have used the most due to its low weight and high portability. Important to note that we have connected an external USB wireless network adapter with the Realtek RTL8187 chipset and 5dBi gain antenna in order to achieve a better scanning frequency. This was dictated by the fact that Microsoft's native wireless API that we were using is highly dependent on the drivers of the network interface and does not guarantee that WlanScan() method will send the actual probe requests as documented in [23]. By using this external USB adapter we were able to achieve a stable scanning frequency one scan per second. Scanning frequency of all the other devices using integrated wireless network adapters were approximately one scan per two or three seconds.

4.1.2 Bluetooth

Because of the fact that we were not able to get the number of working Bluetooth stations – we have decided to use regular mobile phones with Bluetooth connectivity. We simply place the phones in strategic locations where we want our Bluetooth hotspot to be and set it to the "Discoverable mode" so that later when we come close to that location - phone can be discovered by the laptop that we use.

Compared with using Bluetooth stations this approach has one disadvantage – the client (laptop) has to initiate the Bluetooth scanning. This in turn brings some noticeable disadvantages: firstly, it wastes more energy as the client has to scan for available Bluetooth devices on a regular basis. Second - Bluetooth signal tends to interfere with Wi-Fi signal and may have a negative effect on scanning available wireless access point signal strengths. This is especially noticeable when using a laptop with wireless network and Bluetooth adapters which are both internal.

Due to the limitation of available resources - this approach was the only way that we could integrate Wi-Fi and Bluetooth infrastructures into one system. In order to minimize Bluetooth and Wi-Fi interference we have purchased an external Class 2 mini Bluetooth dongle. Class 2 Bluetooth devices have detection range of about ten meters - that is still too large area for our needs. In our case – Class 3 Bluetooth dongle with approximate operational range with one meter would have been a better solution, however we were not able to get any of Class 3 Bluetooth devices at the time being so we had to find a way of how to reduce the scanning range of our dongle.

To be able to control the scanning range and further reduce Bluetooth and Wi-Fi interference we have experimented with wrapping the dongle and Bluetooth phones into aluminum foil as illustrated in figure 4.1.

Figure 4.1: Mobile phone wrapped into aluminum foil

Using this approach we were able to reduce scanning range of Bluetooth dongle down to a few meters and that also helped to minimize the negative effect that the Bluetooth had on our wireless network adapter.

By using Microsoft Windows Bluetooth stack we were able to achieve a scanning frequency of approximately one scan per three seconds.

More analysis about Wi-Fi and Bluetooth signal interference is presented in section 4.7.7.1.

4.2 Environment

This chapter describes the environment where experiments, development and testing of the indoor positioning system took place. It is very important to mention the environment that the system was evaluated at because the indoor positioning system could demonstrate completely different results while being deployed elsewhere. There are a number of factors that have a high impact on the accuracy of the system such as deployment of the access points (position-wise), total number of access points available and the stability of signal strengths throughout the day.

4.2.1 Access Points

We were developing, deploying and evaluating the system in the department of computer science of Aalborg University.

The total number of physical access points that are deployed throughout the area is around 35. Figures A.1 and A.2 located in appendix A illustrate the deployment and radio coverage of all the access points on the ground and first floor.

As it can be observed – the upper part of the building has a much better radio coverage compared with the ground floor. It is important to note that we were not able to change the positions of these access points in any way so the topic of deploying the available access points is not within the scope of this project.

There are two different models of access points that are currently installed: AIR-AP1131AG-E-K9 and AIR-LAP1142N-E-K9. Both models belong to the same Cisco Aironet family and share similar characteristics.

The unique feature of these access point models is that they are able to emit the radio signal on three different channels simultaneously. In our case – each of the access points were operating on channel 1, 6 and 11 at the same time. While operating in such mode access point assigns a virtual Media Access Control (MAC) address for each channel. This means that our scanning devices were able to identify three different signals that were coming from a single access point. On average – in each position we were able to collect approximately 40-65 different MAC addresses even though there were only 35 physical access points available in total.

4.2.2 Centralized wireless control system

Another very important factor related to the wireless network deployment in the department of computer science of Aalborg University is the usage of centralized wireless control system. By using above mentioned Cisco access point models it is possible to connect them all to one central device called Wireless Network Controller (WNC) which can later be used to control the number different parameters of each access point. According to [4] it makes it easier to manage such large scale access point deployments as the radio coverage and signal strength of each access point can be dynamically changed to better adapt current requirements.

Using this approach the centralized system can automatically increase emitting radio signal strength of each access point when the load increases. Similarly – the system might decrease signal strength or completely power off some of the access points where the radio coverage is good enough. According to the description from Cisco [5]: when access point can detect three or more different access point signals with power levels greater than -65 dBm – it assumes that the area is already covered pretty well and it does not operate on full power.

Figure 4.2 illustrates the typical signal strength fluctuations during the whole day. The measurements were taken with the same device that was positioned in our group room and measured signal strengths from all the available access points every minute for 24 hours.

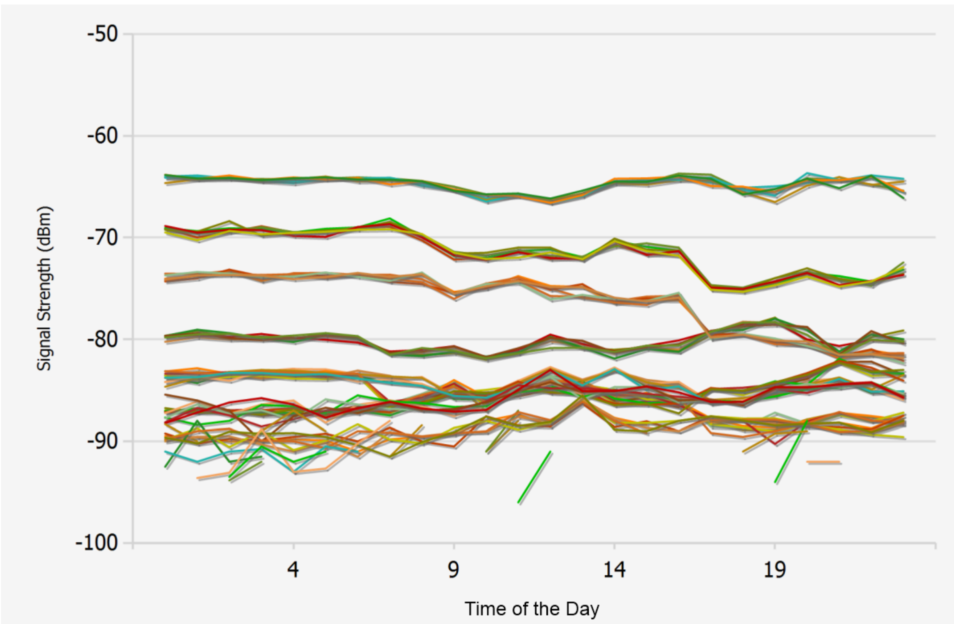


Figure 4.2: Signal strengths obtained from same position

Figure 4.2 very clearly demonstrates what impact Wireless Network Controller has on the available access point signal strength values. On a typical day there are a lot of wireless network users in the building that at some point connect or disconnect to/from the network. It means that signal strengths from certain access points are constantly changing while trying to adapt to the current situation.

We can observe that Cisco with its new technologies makes it very easy to manipulate and deploy wireless networks – however from indoor positioning point of view – this brings big disadvantages because you cannot expect the signals from different access points to stay persistent all the time. That means that it is not possible to have one radio map that could be used all the time – additional techniques should be employed in order to efficiently use indoor positioning system in such an environment. As mentioned before – one of the goals of this project is to make indoor positioning system adaptable to different situations. This is the reason why this dynamically changing environment suits us very well.

4.3 Measurements

In our experiments we use different types of measurements in order to present results of every experiment. Every existing measurement supplements each other. In general we use three measurement values: accuracy, walking error distance and Euclidean error distance.

$$Accuracy = \frac{correct(estimations)}{all(estimations)}$$

Accuracy is expressed in percentage of correctly estimated positions to all positions which were calculated using Euclidean distance between values of the signals, received from the same access point. Error distance is a metric distance between true position and estimated position. We present two types of error distances: walking and Euclidean, which are expressed in meters. Euclidean distance is computed based on Euclidean distance formula 3.1 between coordinates of each position. As we have used raster map in our application pixels were employed to compute distance between positions and converted to metric measurements. Knowing approximate longitude and latitude coordinates of the map corners we have expressed the width of the building in metric form. Lately this value was associated with amount of pixels representing the same width of the building on the raster map.

However as we are dealing with building topology, shortest walking distance is usually bigger than Euclidean distance. In the building topological constraints exists which not allow people to go through the wall, and jump through the corner. As in our system topological constraints are expressed by graph model, walking distance is computed by shortest distance in the graph. In other words node in the graph can be reached by other nodes only through available edges. To compute shortest path in the graph we use simple breadth-first search (BFS) algorithm. Search is performed by exploring neighboring nodes. If exploring node is not the one you are searching for, then process continues visiting unexplored neighbors of recently checked nodes. In the worst case complexity of computing shortest path would be $O(|V|+|E|)$, where V – vertices, E – edges. SQL function is developed which recursively crawls connections table in order to find destination node.

In order to save computational expenses we take average scale between positions, which in our system is approximately 4 meters. Walking distance measurements are approximated. In almost all cases walking distance is longer than Euclidean. However in situations where actual distance between positions is higher than 4 meters, Euclidean distance might be longer.

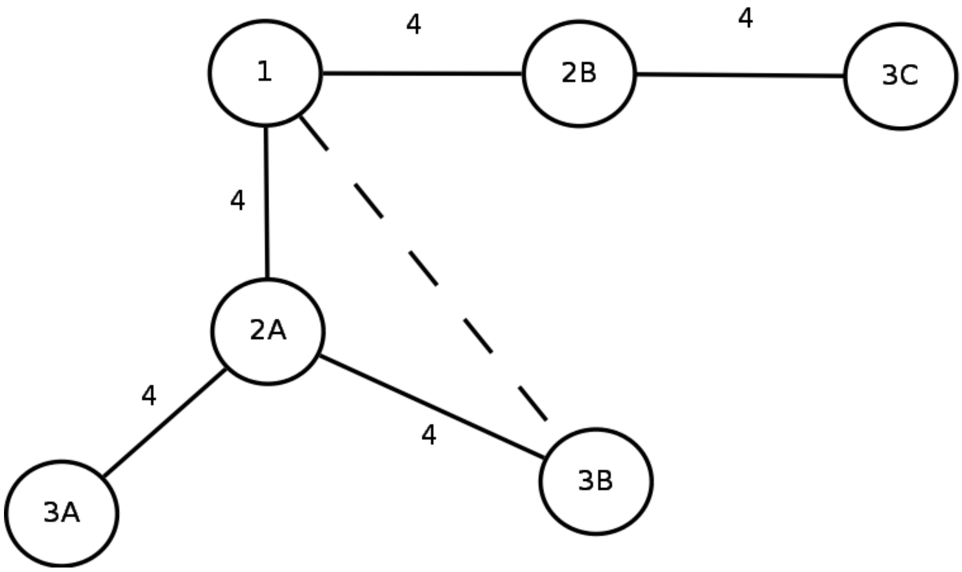


Figure 4.3: Distance comparison

In figure 4.3 we can see graphical presentation of two metric measurements and their difference. Let say, our current location is position 1 and our destination position is 3B. BFS first will check nodes 2A, 2B. Having no luck it will search in the 3rd level: 3A, 3B, 3C. At this moment BFS stops and it returns result 8. As every edge has the same value 4 and there are only 2 edges separating destination and current position. Euclidean distance is shown by dotted line directly connected with 1 and 3B. This distance is definitely smaller than walking one.

In every case where distance diagrams are presented, average error distance is included. Only Euclidean average distance is demonstrated, as it is more precisely computed comparing to walking distance.

There are few time-based experiments, where performance of some operation is compared. In that case we use seconds to measure performance.

4.4 Types of experiments

In our project we perform two types of experiments: simulated and real life experiments. In every type of experiment we use real signal data collected by computer – it is not generated in any case. Online phase however differs between both cases. In real life case we physically walk and collect current signals from the access points, in simulation, movement of the object is generated and imitated.

4.4.1 Simulation

We employ simulation type experiments to test our most cases. Because of the huge amount of experiments we need to run and limited time we have to evaluate our system, this type of the test gives us more flexibility and control of. We can run it from one place and one computer – no physical movement is required. Moreover tests are performed few times, which means it stabilizes the output results. It also has time-saving value.

For simulation type of experiment we use data collected only in offline phase. Later this collection is divided into two parts: test signals and training signals. Test signals are used as an input of the experiment, which is evaluated based on training signals. This idea of test and training division was taken from data mining, especially classification process.

We have developed procedure *divideSignalsToTestTrain* which divides offline phase signal data into not overlapping subsets test and training data. We have raised the problem to select signals not totally randomly and accidentally, but keeping ratio between test and training data in every position which was fingerprinted on the map.

Parameters in this procedure are: *testRatio* – ratio between test and training data, *lHBound* and *uHBound* are lower and upper hour bounds used to separate data into two time zone regions and only then apply the ratio. Actually ratio is kept not between amount of signals but between amount of scans performed in one position. Every scan contains stamp of AP "seen" at the particular time and it has atomic value. Taking some signals from one scan and other signals from the other scan will distort evaluation results. 90% ratio is used as a default, as far less data is required for testing purposes. For example, if in one position 150 scans were made: having 90% ratio 135 scans are stored in training signals table, and 25 scans – in test signals. Test ratio is kept also between scans made in different time zones. If in one position 180 scans were made: 100 scans in the morning and 80 in the evening having such lower and upper hour bounds which precisely divide the day into morning and evening – the output would be 90 scans stored in training signals and 10 scans stored in test signals as first time zone, and 72 scans stored in training signals and 8 scans stored in test signals as second time zone. After having different time zone signals with preserved scans ratio we can perform time based experiments.

After data is prepared to invoke movement in the system routes must be constructed. We have proposed three user generated routes. Routes are generated based on most common paths we tend to follow. Route predefines sequence of true positions and we ensure that object will visit them.

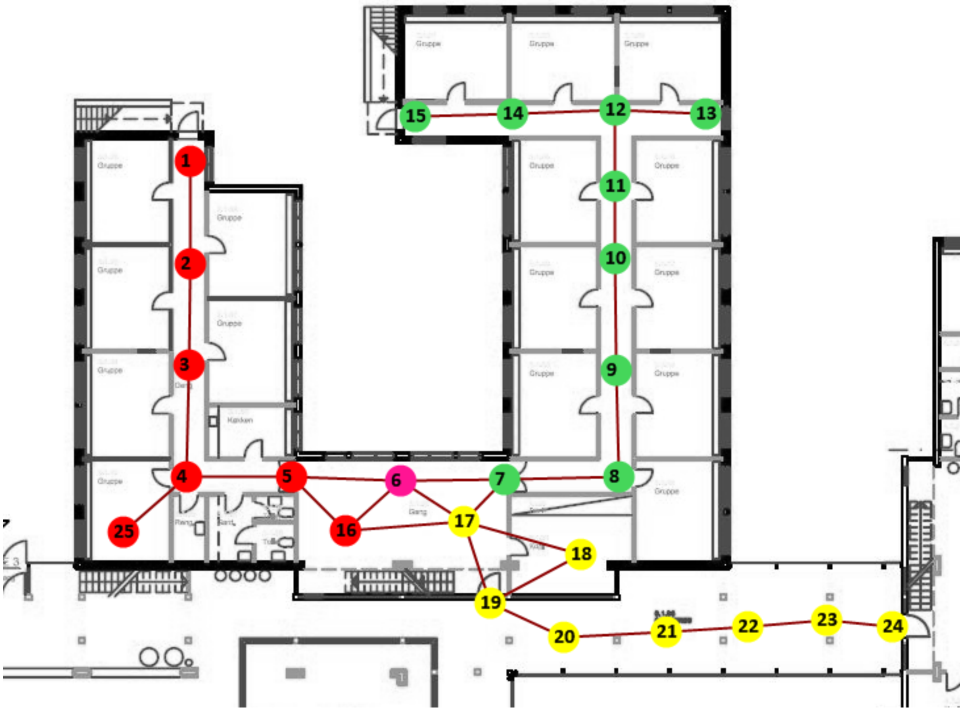


Figure 4.4: User defined routes used in simulation

In the picture 4.4 three routes are presented. Every route is connected together in position 6. It allows running tests using one route back and forth and then smoothly selecting other route.

Route color	Sequence
Red	6, 5, 4, 25, 3, 2, 1, 2, 3, 4, 5, 16, 6
Green	6, 7, 8, 9, 10, 11, 12, 13, 12, 14, 15, 14, 12, 11, 10, 9, 8, 7, 6
Yellow	6, 17, 18, 19, 20, 21, 22, 23, 24, 23, 22, 21, 20, 19, 18, 17

Table 4.2: Routes

Direction in every route is documented in the following table 4.2 . These routes do not change - they are static during all experimental process.

Algorithm 2: Simulation steps

```
Randomly select the route;
repeat
    Randomly select signals from online signals table of a single scan's
    ;
    Estimate position based on train signals;
    Record results in poshistory table;
until 30 routes are visited ;
Results are evaluated;
```

Every simulation follows steps presented in algorithm 2 Firstly one of the routes from table 4.2 is chosen. Route is needed in order to simulate true positions of the moving object. If position is 6 then random scan results from test signals is selected and based on this signals position is estimated. In the repeat-until loop available date is randomly selected from test signals dates. True position and estimated position is saved in *poshistory* table. After loop results are collected from *poshistory* table and based on true and estimated position history accuracy and distance error is computed.

4.4.2 Real life experiments

Few cases in our project were tested in a real situation. We have run these experiments in order to check how much use we can actually get from the system we have developed and how much results are different comparing to simulation results. In real life cases we are walking using similar paths as showed in the table. We move and scan simultaneously; in reference locations we stay put for a while. Scanning is performed with Realtek RTL8187L NIC.

It is important to mention that position is estimated based on fingerprints collected few weeks ago. It means that results may be not as accurate as if we would have tested it right away after offline phase. Moreover there are factors of humidity, different flow of moving people, motion of the current user, interference, which can have a negative impact on signal reception, consequently possible deterioration in position estimation accuracy. Simulation is much more isolated from these factors.

Real life experiment is executed in similar way as simulation 2. However because of flexibility reasons route is not predefined for user - it is dynamically composed by testing user (tester) during his walk. If there is predefined route and fixed speed set, tester is much more constrained and does not have much control of the movement in case some technical problems occur. True position is determined by user – position is selected on the map. In situation when current location is between two reference points, user chooses point which is closer to the current position. This way of running experiments brings some error to evaluation results, as user can change position not on time. Comparing to simulation real life experiments are less precise. They are more complicated to be evaluated very accurately.

All signals in database are used to estimate tester's position instead of training signals only. Division into test and training signals in real life case is not needful and not used.

4.5 Data Collection

In this chapter we describe what steps were taken before we deployed the system for experiments and how data (location fingerprints) were collected.

Before the actual data collection step there were a few tests accomplished in order to specify the optimal resolution of reference points in the grid and to select decent quantity of scans in every position.

In the paper [12] authors present suggested values for grid spacing in WLAN indoor positioning system based on the mathematical model. According to them - the optimal distance should be higher than 1.25 meters. However due to specific environment we are running our experiments in, where signal strengths variations are frequent during the day, we calibrate the optimal resolution and quantity of scanning samples in one position for our environment in experimental way.

4.5.1 Determining the optimal density of reference points

It is very important to decently define the density (resolution) of points in radio map. Inappropriately defined resolution might lead position estimation to have high error distances and low accuracy. This happens in case when spacing is small between positions. Position estimation might become very poor, because every position is close to each other and their signals are similar. Not enough information exists to distinguish them and predict location correctly. On the other hand radio map with low density does not give much information about moving object and tracking loses his value.

There is actually no guideline available on which optimal resolution of the radio map should be chosen. As authors claim in [12] grid spacing is not uniform for every system, because of different building topology and different amount of APs. Therefore we have made experiments by varying distance between positions and then logging accuracy results.

In resolution determination experiment we employed only two positions. Distance between positions was minimized in every following case until results did not satisfy our expectations. In every position we have collected 50 signal vector samples. Data was collected and tested with Realtek RTL8187L NIC. Scanning was made with 1Hz frequency. After offline phase signal collection we have done testing with 200 scanning samples. Based on this results accuracy and error distance is computed.

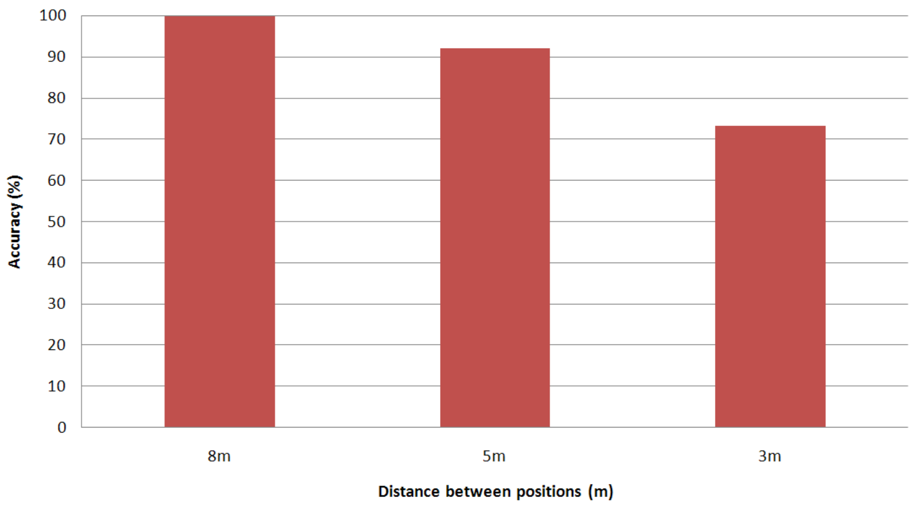


Figure 4.5: Accuracy of position spacing calibration in radio map

In figure 4.5 we can see results of resolution calibration. It is clearly shown that correctly estimated position accuracy drops down when the distance between two points is lowered. It can be explained by the fact, that reducing spacing between reference points make fingerprints more similar thus position estimation less accurate. With 8 meters spacing we have achieved accuracy 100 %, with 5 meters - 92.16 % and with 3 meters – 73 %. We have assumed that reducing distance even more, it would negatively affect accuracy, which does not satisfy our intentions. Furthermore due to changing nature of signals in our environment we have not considered distances lower than 3 meters.

The same tendency of results only inversely proportional is noticed also among error distances. In figure 4.6 with 8 meters spacing walking and error distance is equal 0, as all positions were estimated correctly. With 5 meters resolution we achieve 0.39 meters of walking and 0.36 meters of Euclidean error distance, while with 3 meters spacing - 0.93 meters of error distance for both types. In one case distances differs because for walking distance computation we use average resolution value, in other words not very accurate. For example we set 5 meters between nodes as a walking distance, while in real Euclidean distance based on coordinates might be 5.1 meters. From figure 4.6 3 meter error distance 0.93 still can be applicable, however it worth to mention that this test was run in nearly ideal environment: online phase follows the offline phase right after, tester is not moving and only signals in two positions are considered. More positions can reduce accuracy, as some of the signal vector samples due to negative factors can be similar at particular time.

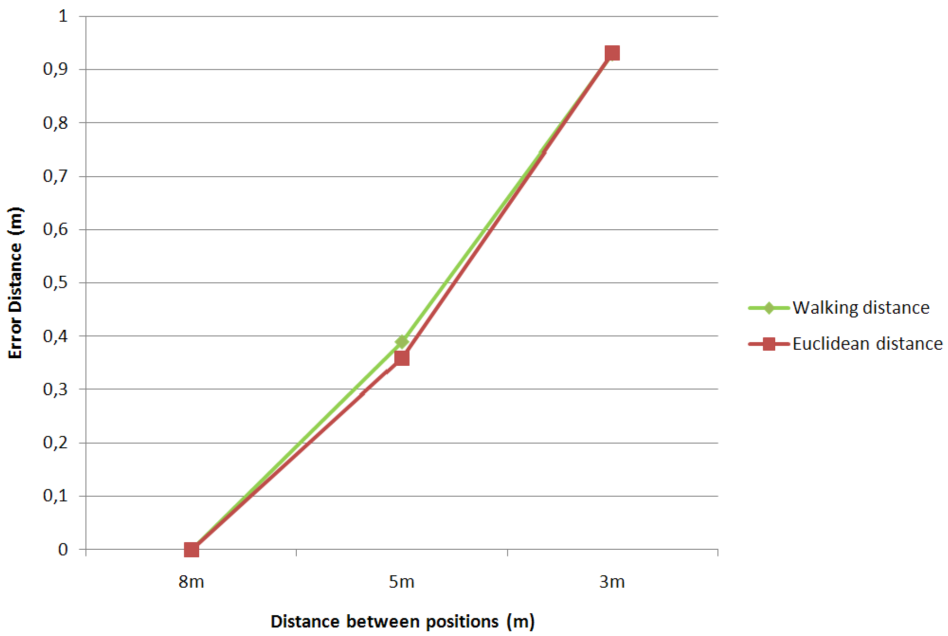


Figure 4.6: Distance error of position spacing calibration in radio map

Considering facts about environment, as a consequence we have chosen approximately 4 meters of spacing in the radio map where still we can reach high accuracy and low distance error. But it does not mean that this resolution is kept in every place throughout the all radio map. Spacing varies between 4 up to 5 meters in the all radio map. On the other hand distance of 8 meters is too extended to make precise tracking in the system. That is why we ended up with 4.42 meters of the average resolution in the radio map.

4.5.2 Quantity of scans per position

Before populating radio map with defined pace of 4 meters we have faced the uncertainty problem related to amount of scans per position we have to perform in order to keep position estimation accuracy high. Our expectation was that small quantity of scans collected during little amount of time can be influenced by negative Wi-Fi factors or simply some access points can be not seen during short time and suddenly become visible [12]. Therefore we assume that with small amount of scans average signal vector value is not settled down.

In order to find the reasonable number of scans we have run few tests. Similarly to radio map spacing tests we have two fixed positions where each position was populated with signals. Distance between reference points was set to 4 meters based on results from previous test. Amount of scans in offline phase was varied from 5 scans up to 50 scans. We have not considered interval higher than upper bound of 50 scans due to limited time we had. Hereinafter testing is done from one position. Conclusion is made based on 200 samples collected in every test. Data was collected and tested with Realtek RTL8187L NIC.

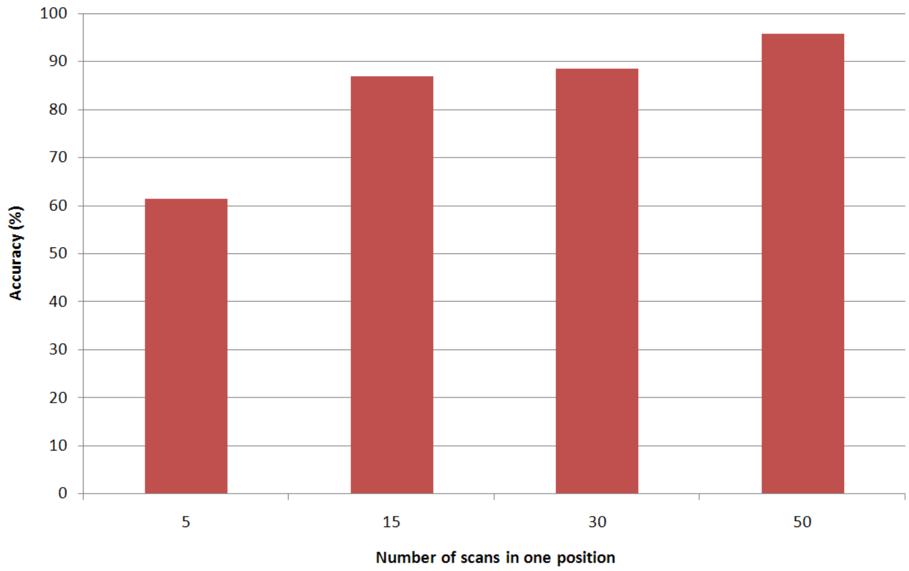


Figure 4.7: Quantity of scans per position

The outcome confirmed our expectations. As shown in figure 4.7 incrementing quantity of scans results in the growth of accuracy of correct estimations. Accuracy varies from 61% up to 95.8%. With value of 15 and 30 scans accuracy difference is only 1.6%. Between 30 and 50 scans variation is 7.2%. Difference between 15, 30 and 50 scans are not so significant comparing with 5 scans. We could use 15 scans instead of 30 or 50. However we have decided to collect signal data by scanning every reference point 50 times. We claim that more signals means more stabilized average of signal strengths. Furthermore tests are performed right after the offline phase, which means that signal fluctuation is not that visible comparing to the fluctuation which can be seen after longer time. That can affect system performance. Collecting only 15 scans with 1Hz frequency some signals may not be visible at that particular moment. 15 scans comparing to 50 scans guarantee less stabilization.

These accuracy values are collected in the environment with two positions. Expanding radio map can affect the accuracy and error distance.

4.5.3 Collecting the radio-map

After we have determined the ideal distance between two reference points and what is the optimal number of scans in each point we then started to collect the radio-map for the area where the experiments took place in the later phases.

In total – we have collected position fingerprints from 25 different reference points covering the building segment that our group room is located in. Figure 4.8 illustrates the building area and where each reference point is located.

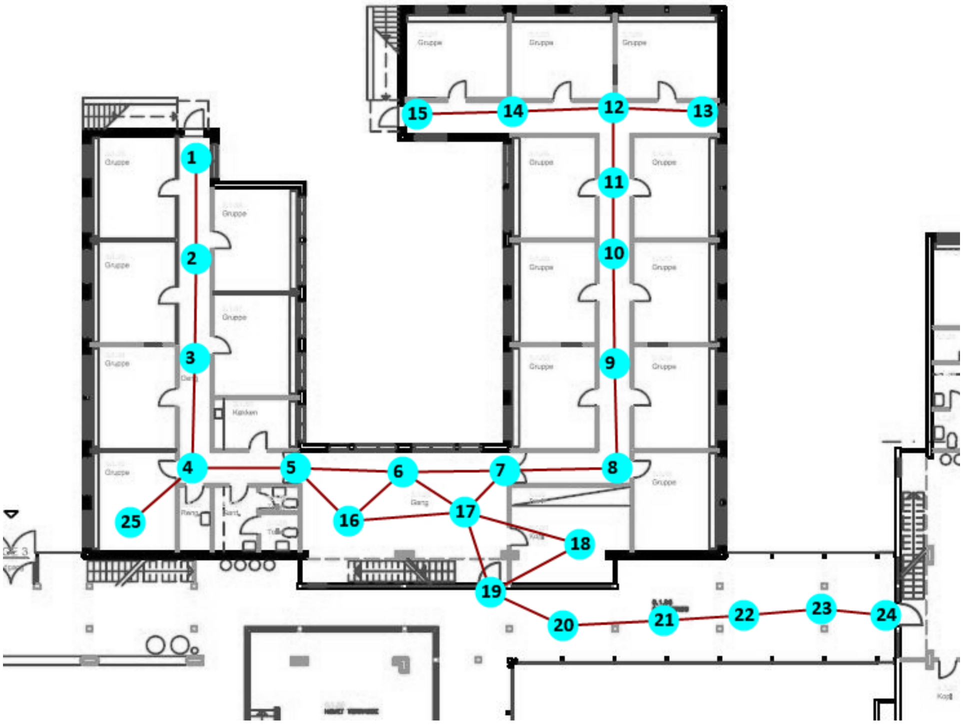


Figure 4.8: Positions of reference points

Two laptops - Asus EeePC 901 and HP Compaq nc6000, with different NICs were used to collect the fingerprints in each reference point. To form the radio-map that was used in our experiments we made 50 scans with both laptops in each position. Two other laptops were not used to collect location fingerprints intentionally as we had some experiment cases that required a user not to have his fingerprints in the database.

Additionally, while conducting some experiment cases we needed to have several fingerprints that were collected during different time of the day. That is the reason why we have collected fingerprints from all the available reference points with Asus EeePC twice - once in the morning and later in the evening. Signal strength values tend to be more stable during the evening as there are very few users that might cause signal strength fluctuation.

Furthermore – in order to run direction prediction experiment we needed to collect fingerprints from each reference point each time facing different direction. We then collected additional data in each reference point while facing east, west, south and north – ten scans per direction.

To sum it all up – the fingerprint of each reference point in our system is comprised of 190 scan results in total.

It took us two days to collect the whole radio-map that later was used when carrying out the experiments. The whole data was collected approximately one week before we started to run and evaluate the experiments.

4.6 Parameter determination

Our positioning system depends on few parameters which have an impact on position estimation accuracy. Before running tests on specific cases our task was to set these parameters. Each constraint was NOT randomly chosen. Determination was performed in experimental way - in other words varying and adjusting the value of parameter which gives the highest results. Selected parameters were used during all experimental cases. Penalty and filter parameters are described in the following sections.

4.6.1 Penalty

In position estimation method 1 best position is determined by computing distances between reference point average signal strengths (SS) and online scan signal strengths. Every signal sent from access point is associated with unique MAC address. So in other words we can say that distance is calculated between identical MACs' SS. To make it clear, overlapping MACs are sought between the set of MACs from single online scanning result set and from single specified fingerprint. There are no MAC duplicates in these sets. Continuing, let suppose, in situation of different MACs no action is taken. Finally best reference point is chosen based on the closest and most similar fingerprint stored in database.

During the testing part we have noticed that in case of different MACs some additional actions should be taken as some of position because of this situation might be predicted incorrectly. We employ penalty. In the following paragraph the motivation and value of penalty is presented based on a simple example.

	FINGERPRINT 1	FINGERPRINT 2	ONLINE SCAN
MAC1	-50	-50	-50
MAC2	-	-61	-60
MAC3	-	-70	-70
MAC4	-60	-60	-60
MAC5	-50	-	-
MAC6	-70	-	-
MAC7	-50	-	-
MAC8	-	-80	-
MAC9	-	-90	-

Table 4.3: Example - SS vectors

Let consider that we have two fingerprints as demonstrated in the table 4.3: FINGERPRINT 1, FINGERPRINT 2.Let say we have 9 access points associated with unique single MACs. Based on quantity of scans experiment 50 scans are made in one reference point. Therefore duplicate MACs exist. Identical MACs are grouped together and average of SS is calculated. Suppose, tester is in certain position on the map and once he has scanned for Wi-Fi signals. In one scan only unique MACs exist. No average is calculated. Those MACs with SS are presented in the column ONLINE SCAN.

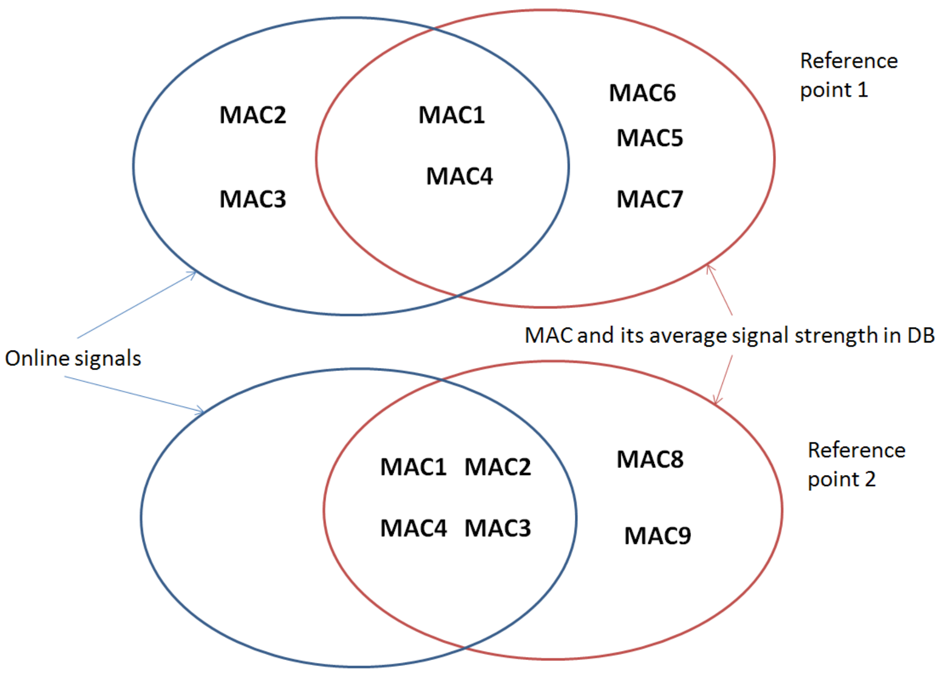


Figure 4.9: Example - overlapping MACs

In the figure 4.9 is visually demonstrated overlapping between similar MACs comparing ONLINE SCAN with FINGERPRINT 1 and ONLINE SCAN with FINGERPRINT 2. Blue ring in two cases represent the same scanning result set. Regarding SS indicated in table 4.3 almost all overlapping MACs share the same values, except MAC2 in FINGERPRINT 2. Intuitively it seems that reference point 2 has to be the outcome of estimation as there is larger set of visible overlapping MACs with almost the same SS value. However after computing the distances with Euclidean formula, the outcome is not expected:

$$Dist(ONLINESCAN, FINGERPRINT1) = 0$$

$$Dist(ONLINESCAN, FINGERPRINT2) = 1$$

FINGERPRINT 1 is closer to ONLINE SCAN because of the small difference of SS in FINGERPRINT 2. In order to avoid such situations penalty for not overlapping MACs have to be used. In this case underweight must be added to MAC2 and MAC3 in the first case. Underweight is added only to the nonexistent MACs in the particular fingerprint.

Almost in all cases amount of set of fingerprint's MACs is bigger than set of MACs from online single scan. That's because one fingerprint usually contains more than 50 scans and probability to collect more MACs is definitely higher. Moreover fingerprints are often collected statically with NIC which have a good specification and higher coverage range. Applying penalty for nonexistent MACs in online single scan may negatively influence distance calculation.

Intuitively penalty must depend on SS from online scan. If signal is strong then penalty must be higher, if signal is weak penalty must be lower. That's why the constant PENALTY in the algorithm 1 must be elaborated:

$$PENALTY = (SS + penaltyValue)^2$$

SS is signal strength of MAC which is absent among fingerprint MACs. Parameter *penaltyValue* is value of penalty which was changed and added for such absent MACs in order to overweight the Euclidean distance between fingerprint and online signal vector.

We have run few simulation tests with default configuration (random 10 routes) with different values of penalty 4.10. According to the scale of possible SS [-50;-100] we have chosen step values from 50 and 100. Penalty 500 was selected in order to test if very high penalty can overestimate result.

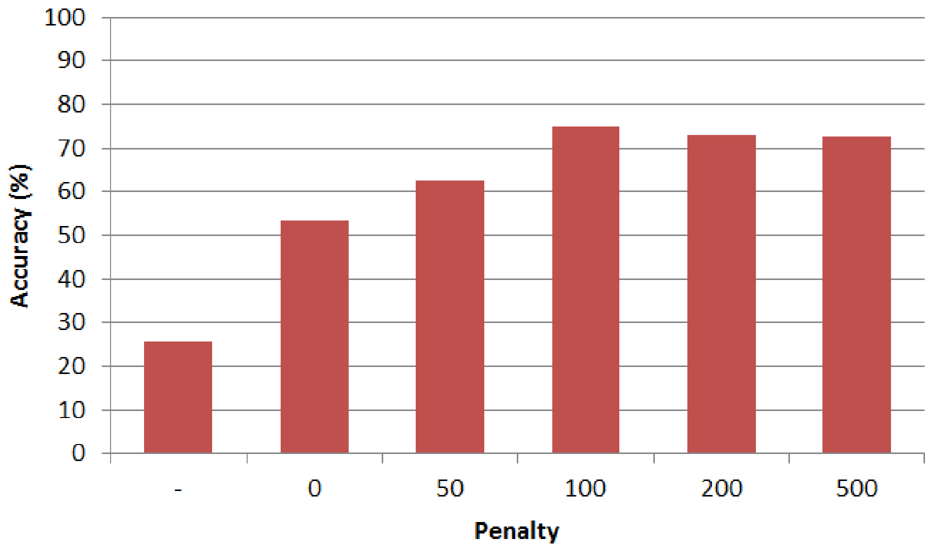


Figure 4.10: Estimation accuracy achieved with different penalties

As we expected, results in the figure 4.10 where penalty was ignored (certain MACs from online scan do not exists in a fingerprint. Marked with "-" in the diagram), brought the worst accuracy among the others. It demonstrated very low 25.9% of estimation accuracy. The highest accuracy was achieved with penalty 100 – 75%. Only small fluctuation is noticed in the last three cases with penalty 100, 200, 500. It seems that adding higher penalty than 100 has not much meaning. Results with penalty value 100 differ with penalty 200 only by 1.95% and with 500 only by 2.2%.

Same tendency only inversely proportionally is noticed in distance measurements in the figure 4.11. Error distance is almost not acceptable in case of penalty value 0 and no penalty. Distance values for no penalty reaches 8.86 meters (Euclidean) and 12.09 meters (walking), while results with penalty 0 present more reasonable errors of 3.87 meters (Euclidean) and 6.46 meters (walking). There is a huge drop between no penalty and penalty 100 – 7.45 meters (Euclidean) and 10.08 meters (walking). This shows about importance of the penalty and success of the system's performance.

Error distances with penalty equal or higher than 100 are almost stable as shown in the figure 4.11. Applying higher penalty than 100 has almost no impact on error distance: Euclidean error distance variation is equal to 0.2 meters, while walking error distance variation is only 0.01 meters.

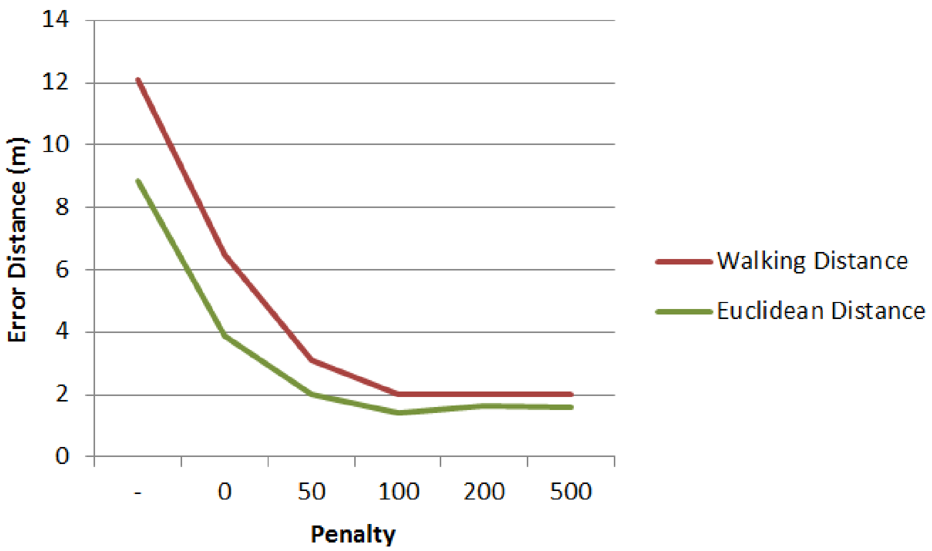


Figure 4.11: Error distance achieved with different penalties

Because of the highest results achieved we have chosen penalty value 100 to use in all other experiments. Values higher than 100 were not considered as their outcome was not significantly different.

4.6.2 Filtering noisy signals

In this subsection weak signal filter is presented. It is explained that parameter value need to be carefully selected, as too high value can lead to the loss of vital information and consequently loss of the precision of computation. Correctly chosen value can improve accuracy of position estimation.

After the long term scanning we have detected interesting feature about weak signals. Signals with very low strength tend to be very unstable, chaotic and even random. It means that you cannot expect good outcome from these signals. Some of them can slightly distort correct position estimation. Our goal was to find the universal optimal value filter for every NIC we used and to test out if removing signals below this filter during offline and online phase can guide system to better results.

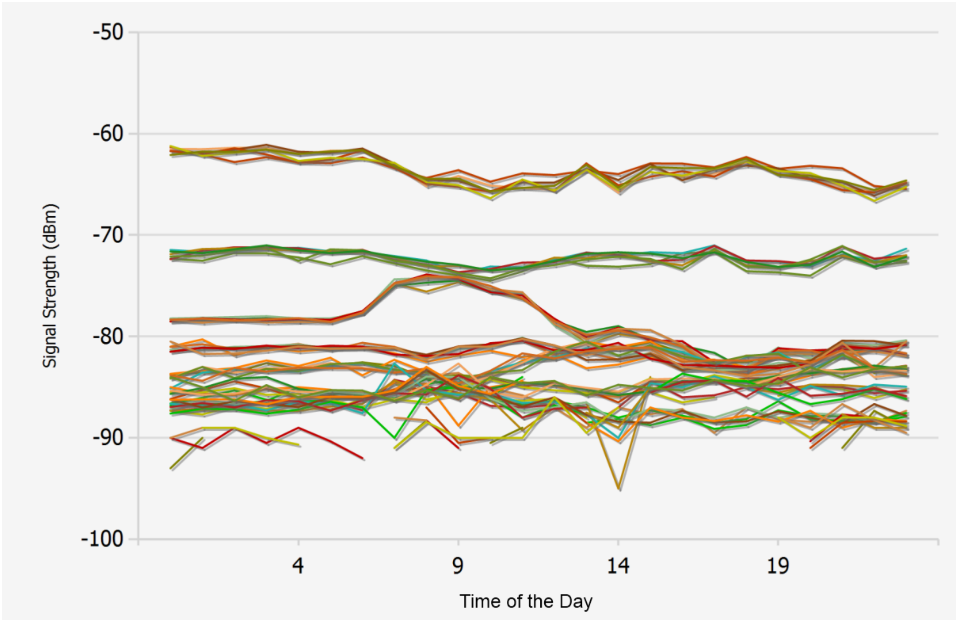


Figure 4.12: Signal strength variation during the day

In the picture 4.12 fluctuation of the signals during whole casual day is demonstrated. This behavior is also noticeable in figure 4.2Scanning frequency was 1/60 Hz, in other words single scan was launched every minute. All data was collected with HP WLAN W400 network card in one position. From the figure 4.12 it can be noticed that weak signals below around -90 dBm are very disorganized and messy. It might be caused by different negative Wi-Fi factors or just because access point is located far away from the current position. Those signals do not offer much information which can be effectively employed in position estimation.

In order to test this phenomenon we have accomplished experiments where filter is applied. We have run simulation test with 10 randomly selected routes. Realtek RTL8187 NIC was exploited in this test as HP WLAN W400 NIC was running long term experiment. In simulation test and training signals are collected from one recently mentioned device (signals from other devices are not considered), as only filter effect is being checked.

Results of the experiments are presented in figure 4.13. In the first case filter was not applied and accuracy reached 53.9 %. After elimination of signals lower than -90 dBm accuracy was slightly improved by 6.9% comparing to no filter case. Accuracy reached 60.8%. Removing signals lower than -80 dBm accuracy almost have not changed in contrast with the second case. There is almost invisible difference between filter -90 and filter -80, only – 0.6%.

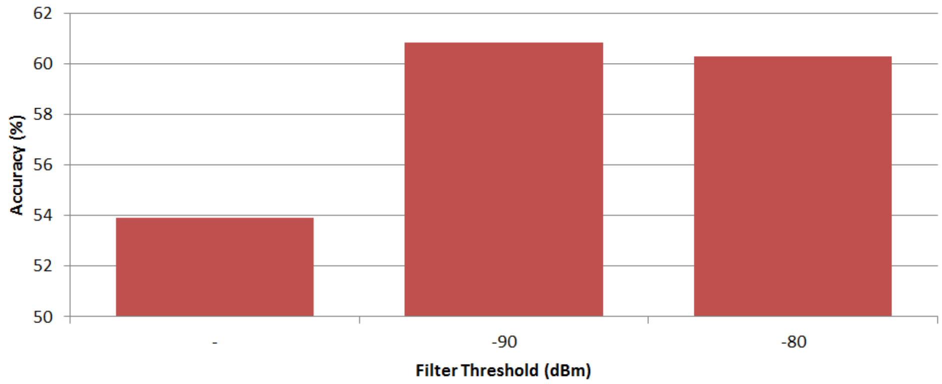


Figure 4.13: Accuracy using different filter bounds

In error distance diagram 4.14the lowest error distance is reached with threshold -90 dBm (lower is better). Error distance drops significantly between first and second cases: from 9.8 meters to 7.93 meters (walking), from 7.59 to 6.35 (Euclidean). Comparing two the next cases walking distance changes only by 0.3 meters while difference of Euclidean reaches 0.8 meters almost the same as between two first cases. Finally to sum it up, with -80 dBm threshold accuracy and error distance is getting slightly worse. It can be explained that with threshold of -80 dBm there are more valuable information. If it is eliminated, accuracy can drop slightly more. Choosing threshold of -80dBm is not recommended to use as a universal value even though the results shows almost equal results comparing to the bound of -90 dBm. In the next paragraph explanation of this recommendation is presented.

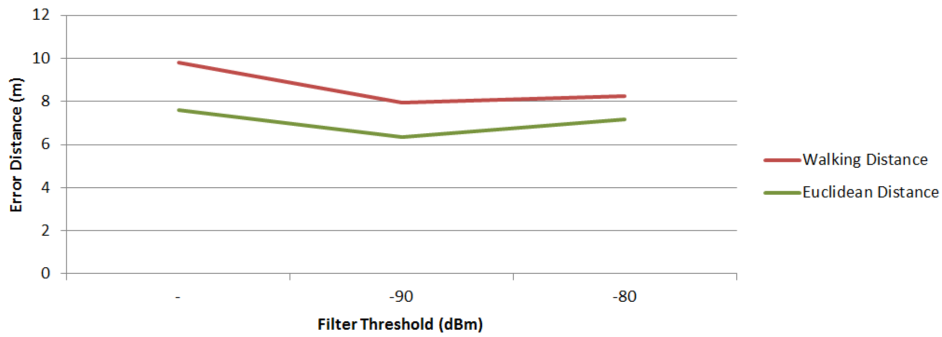


Figure 4.14: Error distance achieved with different filters

If we look at the picture 4.12 more precisely, we will notice that if we apply threshold -80 dBm we will cut almost half of the signal information collected by HP WLAN W400 NIC. It means that with different network cards applying specific filter will receive different results, because different network cards have different bounds for weak signals. High quality NIC in our case Realtek RTL8187 comparing to lower class NIC HP W400 has different type of signal reception. It tends to collect signals with higher strength. We have composed the table 4.4 with percentage of signals with specific strength to show the difference between signal receptions of two cards. These percentages are extracted from radio map data. There are small amount of signals with strength lower than -90 dBm. However as we have assumed if we apply filter lower than -80 dBm in case of HP W400 we will remove almost 40% of important information needed for estimation. It is definitely bad idea to apply filter with threshold -65 dBm in both cases, where more than 10.5% of data exist in the first case and 40.74% in the second.

	Percentage of signals	
Filter bound	Realtek RTL8187	HP W400
-90	0.7	2.4
-80	2.1	37.38
-65	10.5	40.74

Table 4.4: Percentage of signals belonging to specific strength range

As it was mentioned earlier in the picture 4.13 applying filter with a bound of -80 dBm we can notice similar results comparing with situation where threshold of -90 dBm is applied. Recalling to the data in the table 4.4, we can assume that removing 2.4% of noisy signals with -90dBm bound from HP W400 will act similarly as filtering 2.1% data with -80 dBm threshold for Realtek RTL8187. These are only assumptions which were also made based on long term scanning results from HP W400 4.12.

Filter with -90 dBm threshold is chosen to be used in every experiment we have run. There are not much data available in this range for both devices 4.4 and they are considered to be noisy, providing little amount of useful information. Removing those noisy signals can improve system accuracy 4.13. However different network cards exist with different type of signal reception. It means that there is no universal filter value for every network card.

Because of the limited time we had, filtering tests were not performed on the all devices, only with Realtek RTL8187 which was used in most experimental cases. For the future work we suggest to model dynamic filter threshold for every card based on the percentage of signals in specific strength ranges.

4.7 Cases

This section describes all experiment cases we have carried out during the project. Every case explains a particular technique we have applied to deal with specific positioning problem. Results and conclusions of each case are provided as well.

4.7.1 Enforcing physical constraints

Because of the nature of wireless signals and complexity of indoor environment, indoor positioning system can sometimes produce very inaccurate results. Indoor space is usually composed of a variety of objects and obstacles (walls, floors, doors, etc.) which tend to block or reflect wireless radio signals. This is especially noticeable when user starts moving and changes his position constantly. Confused by the unstable reflected signal strength indications the indoor positioning system may estimate a wrong location of the user while he or she is moving.

In order to reflect the relations between different entities in an indoor space, a graph model can be utilized. The graph captures all important physical constraints of a floor plan. By identifying appropriate graph models, we can make the positioning system more robust to positioning errors that correspond to users jumping through walls or traveling big distances in a very short period of time.

In our case each reference point is represented as a graph node. Two neighbor nodes at position *N1* and *N2* are connected by an edge if the user can directly travel from *N1* to *N2* without visiting any of the other nodes. An example of a graph that was used in this experimentation part is depicted in figure 4.8.

Another big advantage of using a graph model is that it also limits the set of possible future locations of a user. When we know the current position and maximum speed of the user – we can assume that within a specific interval of time he or she will only be able to reach graph nodes that are connected by an edge. For example, if we know that a user is currently at node *N1* and we can observe that node *N1* is only connected to node *N2* then the set of all the user's possible locations within a particular time period is $N1 \bigcup N2$.

In the experiments described below we try to experiment by changing the parameter which defines how many connected nodes can be reached from user's current position. We record results of four different cases and later compare it with the case where no graph was used.

Figure 4.15 illustrates the definition of reachable nodes. Node pairs A and B, B and C are connected by an edge. Node A has no direct connection to node C - hence it is not possible to reach node C from node A in one step (when reachability is equal to 1).

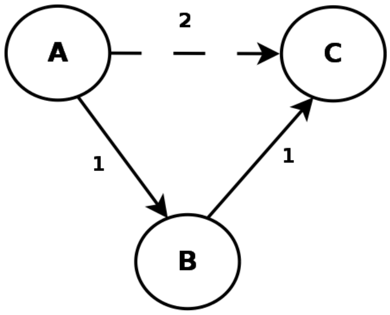


Figure 4.15: Reachability degree

For example, let's assume that user's current position is A. If we set the reachability parameter to 1 it means that user from its current position A is only able to travel one graph edge at a time so he is forced to visit position B before reaching position C. This approach eliminates jumping through nodes and improves performance of the system. If we set this parameter to 2, then user is allowed to move two edges in one step - for example from A to C.

4.7.1.1 Results

Figure 4.16 summarizes the average accuracy of our system that was recorded during this part of experiments. In total – five different tests were carried out. The horizontal axis of the chart represents the reachability parameter where symbol "-" means that no graph structure was used.

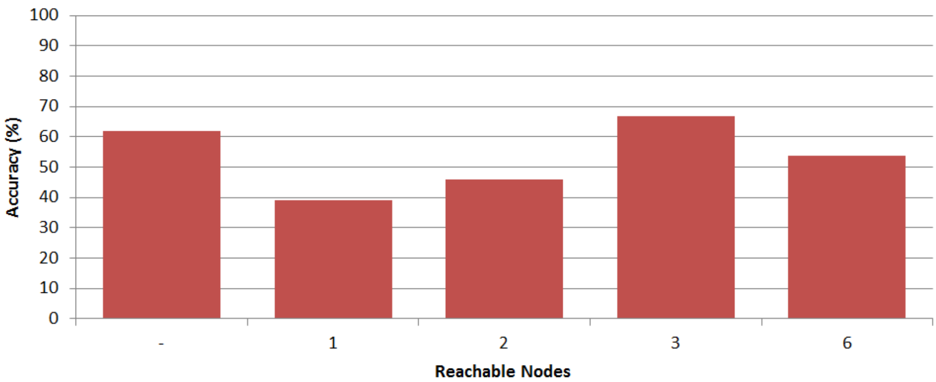


Figure 4.16: Accuracy summary

As it can be observed – the best results were acquired when reachability parameter was set to 3. Using this approach resulted in a slight increase of system's average accuracy compared with the initial case where no graph was used.

Figure 4.17 presents results of the same experiment but in different measures. Here we present the average error of the system that is measured in meters. Two distance measures are recorded: minimal walking distance and Euclidean distance.

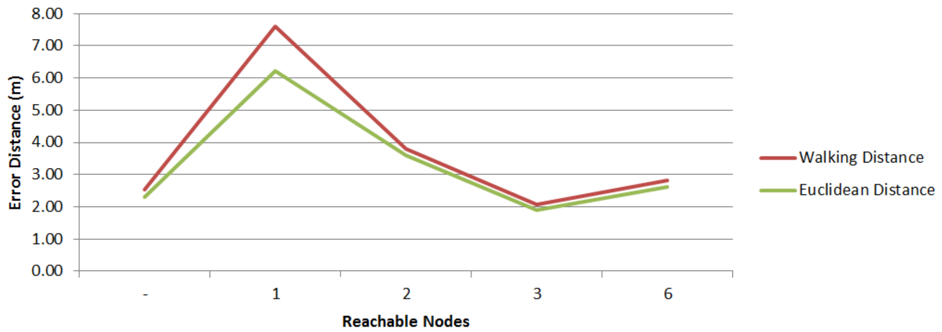


Figure 4.17: Average error distance summary

As it can be seen – system's average error measured in meters has a close relation with the accuracy. As before - case with reachability parameter set to 3 was able to demonstrate the lowest average error.

4.7.1.2 Measuring improvement

Accuracy

To better examine the differences of how graph can change the overall results of indoor positioning system we further investigate results from two cases. Figure 4.18 provides more detailed average accuracy results. This time accuracy is measured in each position separately so that we can better examine strengths and weaknesses of each approach.

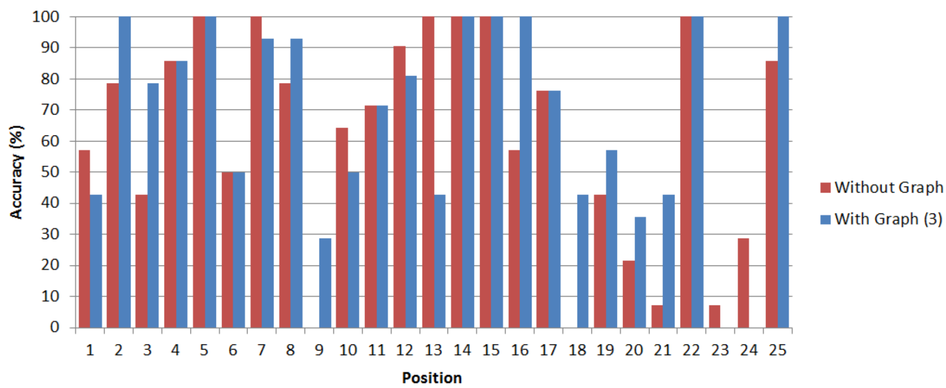


Figure 4.18: Accuracy comparison

As it can be seen - there is no major difference between the two cases in terms of accuracy. In some situations one approach performs better than the other. Overall – by using the graph, system was able to reach average accuracy of 67% compared to 62% that was recorded in the initial case.

Average error distance

Figures 4.19 and 4.20 shows the average error measured in meters for both cases. As previously – average error distance is presented for each position.

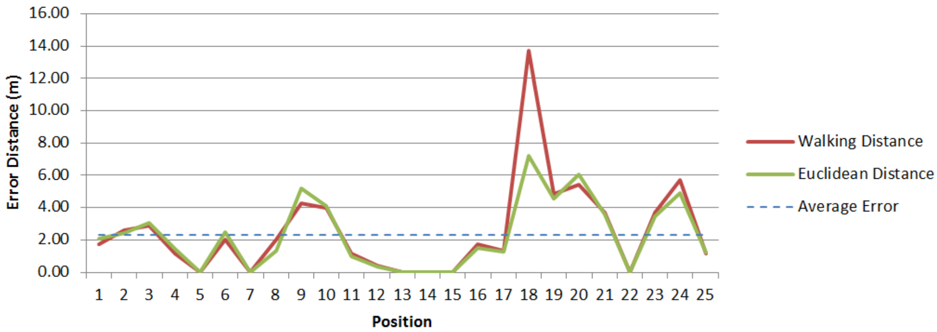
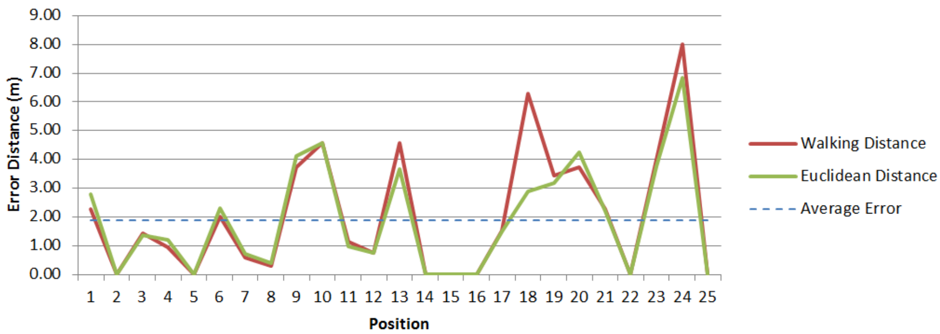


Figure 4.19: Average error distance without graph



graph

Figure 4.20: Average error distance with reachable nodes set to 3

As observed from the figures above - the average error measured in Euclidean distance is again similar in both cases. However, if we compare walking error measures we can observe that the initial approach had a very high error in position 18 which is 13.7 meters. The case with graph was able to demonstrate better average error values. The maximum in this case is 8 meters recorded in position 24.

System performance

A more noticeable advantage of this approach is the performance of the system. By knowing user's current position and changing the reachability parameter we can decrease the set of user's possible positions. That means that the system does not have to consider all the other positions that might not be reachable. As shown in figure 4.21 this might save some computational power and time.

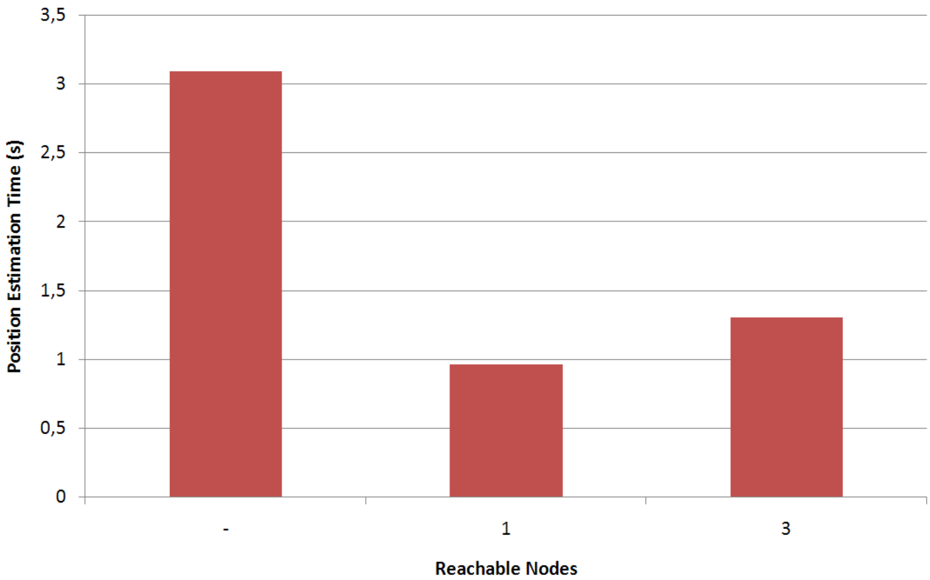


Figure 4.21: Computational time

4.7.1.3 Conclusions

In general – by introducing the graph we were able to increase the accuracy of the system by approximately 5%. The improvement in average error distance was more noticeable: Euclidean error distance was reduced by 17% while walking error distance dropped by 19%. However the main improvement that was achieved while using this method was witnessed in performance of the system. In the optimal case with the best accuracy – algorithm's running time was decreased by 58%. The system has become more than twice as fast as it was before.

It is very important to mention that our experiment was conducted using simulation. In the real world environment the improvement could be different. We would expect a more noticeable improvement in system accuracy and average error distance in a real life experiments. The graph structure should be especially useful in cases where users would start moving. In that case this approach would restrict frequent position estimation errors.

4.7.2 Splitting fingerprints by time

In this experiment section we describe a technique that should make the indoor positioning system more adaptable the environment that we were performing the tests in.

As described in the previous sections of this chapter we are dealing with a very dynamic environment where signal strengths of the access points are constantly changing depending on the current load and many other factors. It is especially noticeable during the peak hours – when there are many wireless network users connecting and disconnecting to/from the network and the load is usually fairly high. During early morning and late evening the signals tend to be more stable. This behavior can be observed from figure 4.2 where most of the signals in the early morning tend to be more stable that during the day.

In order to reduce the negative effect that the signal strength fluctuations may have on the indoor positioning system - we have decided to split our current radio map into two parts according to the time that the fingerprints were taken. That way - one part contains location fingerprints that are recorded during the time interval from 8.00 in the morning up to 17.00 (inclusive). The other part is then comprised of all the fingerprints that were taken from 17.00 and up to 8.00 (inclusive).

Using this approach we can use two different fingerprints depending on what time of the day the user is using the system. If the user is navigating during the working hours we only use location fingerprints that were recorded during that time of the day.

4.7.2.1 Results

To evaluate the results of this approach we have run a test using data collected with Asus EeePC. Fingerprints were collected during a two day period. Once, during the working hours and another time – in the evening. Again – two test runs were conducted in order to compare and find the advantages and disadvantages of this approach over the initial case where no fingerprint splitting was performed.

Accuracy comparison

Figure 4.22 presents the accuracy results of both cases.

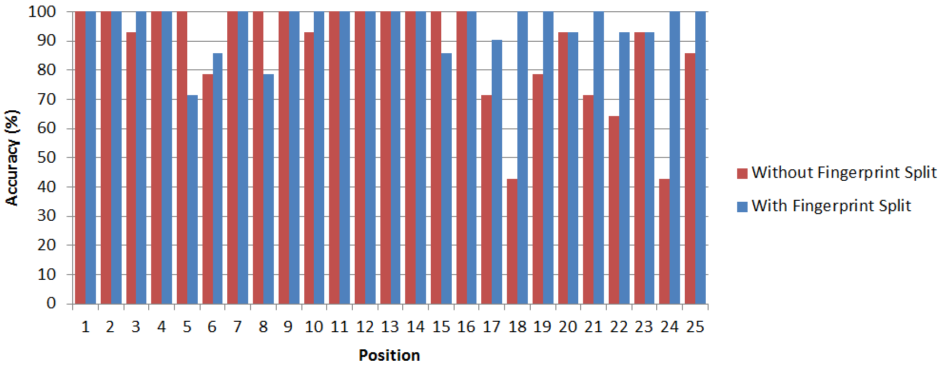


Figure 4.22: Accuracy comparison

Both cases showed very good accuracy. In most positions fingerprint splitting technique was generally a little more accurate. However positions 15, 8 and 5 still have better accuracy in the initial case. The average accuracy in the initial case is 88%. By using fingerprint splitting technique we were able to reach the accuracy of 96% which is rather high measure for an indoor positioning system.

Average error distance

Figure 4.23 shows average error distance of the initial case.

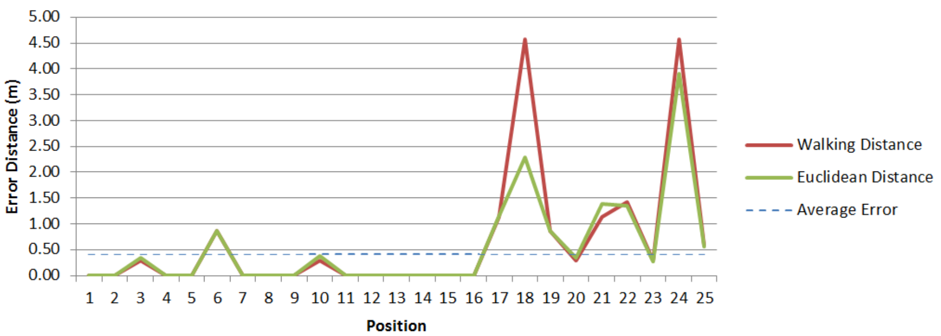


Figure 4.23: Average error distance without splitting

We can easily identify that positions 18 and 24 were the most difficult to deal with for the system. The average Euclidean distance in these positions reached 2.29 and 3.92 meters correspondingly. Average error distance in other positions is more or less minimal. The average Euclidean error distance is 0.55 meters. It tells us that the system was very accurate even without any of the additional techniques.

The results of the case where fingerprint splitting was performed is summarized in figure 4.24.

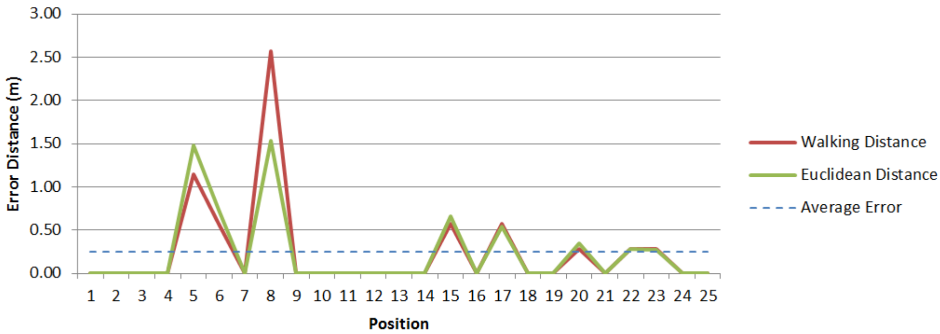


Figure 4.24: Average error distance with splitting

This time the maximum Euclidean error distance is recorded at positions 5 and 8 where it reached the values of 1.48 and 1.54 meters accordingly. The average Euclidean error distance is 0.23 meters. Compared with the initial case – it is a significant improvement.

4.7.2.2 Conclusions

To conclude – we could say that fingerprint splitting is a very effective technique which can both: significantly increase the accuracy and reduce average error distance of the system. Another advantage of this approach is that it does not require doing any of the additional calculations and that is why it does not have a negative effect on the performance of the system.

Overall – by employing this approach we were able to reach an accuracy of 96%. Compared with the initial case – the average error distance was reduced by 58%.

4.7.3 Adapting to different network interfaces

In this experiment are trying to solve a common problem that arises when a user with different configuration – in this case different wireless network adapter, tries to use an indoor positioning system that has its radio map collected by completely different device.

As mentioned in previous sections of this report – different NICs tend to receive wireless signal strength indications slightly differently. It means that if radio map was collected with one type of device – the system's accuracy will usually decrease while working with other devices with heterogeneous configurations.

In order to minimize this negative effect – we make sure that the radio map of our system is collected with at least two different devices. Similarly to the previous case - this enables us to have a few fingerprints in each reference point. Each fingerprint is collected with different device.

Later - when a client connects to the system and starts using it we first measure what fingerprint is the most suitable for his device. In order to do that we need to have the following information:

1. Current position of the client
2. An array of signal strength values that were recorded in that position

To implement this idea we can utilize Bluetooth infrastructure as described in previous section . Whenever the user triggers a Bluetooth hotspot – the system automatically scans the area for available access point signal strengths and uploads this information to the database together with the current position of the user that can be derived from the position of the Bluetooth device.

After having the array of scanned signal strength measures we can compare it with the existing fingerprints that were inserted by different devices. To compare it, we use the same formula as during the position estimation step as described in algorithm 1. We simply measure the Euclidean distance between the two arrays. We then use the fingerprint that has the minimum distance.

Using this method – we can measure how similar user's current device is compared with those which were used to collect fingerprint data. Later – during position estimation phase we only use those fingerprints that are closest and ignore the other. Bluetooth devices enable us to get this information automatically – without requiring any user interaction. It is a good idea to measure the distance of such fingerprints in several locations as this might give more realistic results.

4.7.3.1 Results

This section presents the results of applying this approach.

When conducting this experiment we had our radio map collected by two different devices – Asus EeePC and HP Compaq nc6000. The fingerprint of each reference point comprised scan results of both devices.

To evaluate this approach we have used Compaq nw8440 that had no fingerprints in our database. That explains why the overall accuracy and average error distance measures are not as good as in previous cases.

In order to do a simulation in this experiment – we had to make 10 scans with Compaq nw8440 in each position that our test routes were composed of. We stored those results as test data and used fingerprints from other devices as training data.

Accuracy comparison

Figure 4.25 presents the summary of system's accuracy with and without using this technique.

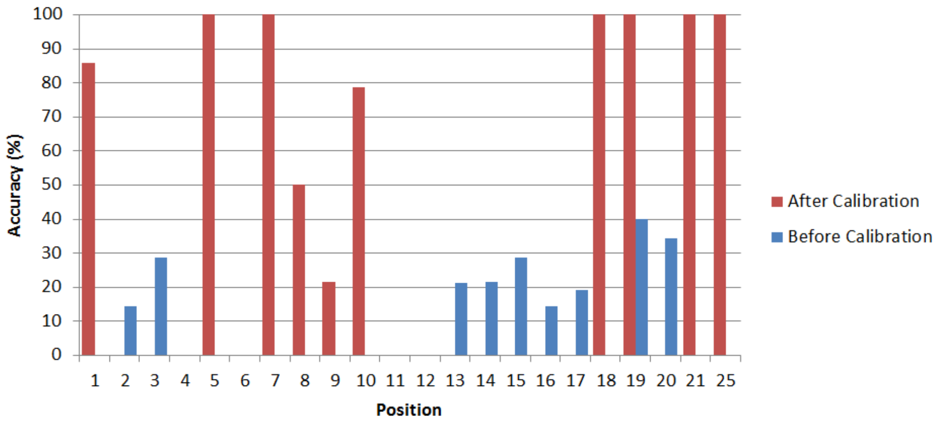


Figure 4.25: Accuracy comparison

As shown in the figure above the results of this experiment are somewhat different from all the other cases. In some positions the system was able to reach better accuracy without using this technique while in others – it improved the accuracy by a relatively large amount. There is only a single position – 19, where neither of both cases had an accuracy of 0%.

The average accuracy among all the positions in the initial case is only 10%. After applying this technique it was increased up to 38% which, compared with the initial case is a noticeable improvement.

Average error distance

Figure 4.26 depicts the average error distance of the initial case.

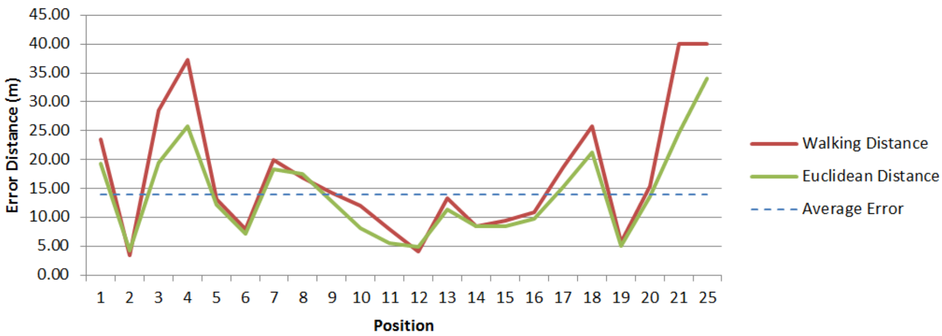


Figure 4.26: Average error distance without calibration

Both - Euclidean and walking error distances tend to vary around the average error distance which in this case is rather high - 13.94 meters. Positions labeled as 4, 21 and 25 were the most difficult to deal with for the system.

Noticeable improvement can be observed by looking at the figure 4.27 which represents average error distance after applying this technique.

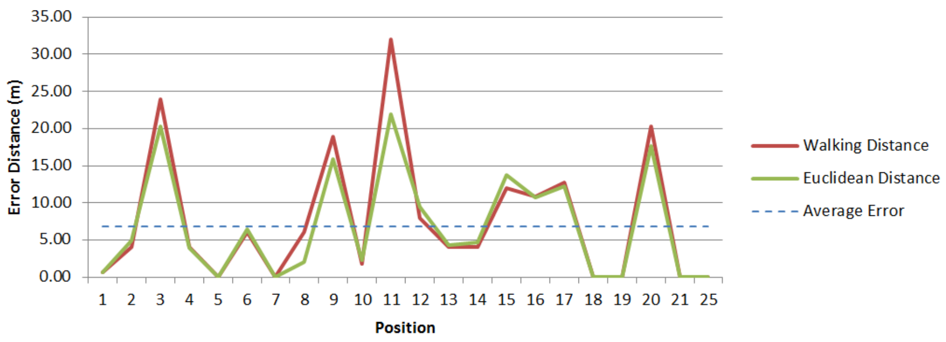


Figure 4.27: Average error distance with calibration

Even though there is no major improvement in every position – the average Euclidean error distance by using this approach was reduced down to 6.86 meters. However there are still some locations that had a rather high error distance. As an example of such positions we can mention the following: 3, 9, 11 and 21. The Euclidean distance reached 21.89 meter mark while walking distance had a maximum of 32 meters.

4.7.3.2 Conclusions

To conclude we can say that this technique does have a positive effect on systems ability to predict user's current position correctly. However, as the results have showed - we cannot expect that the improvement will be achieved in every position.

In general – by employing this technique we were able to increase system's accuracy from 10% to 38%. That might look as a significant achievement, however it still might be too low in order to be used successfully in some particular cases.

The average error distance was also cut to half. Compared with the initial case it was reduced from 13.94 down to 6.86 meters.

4.7.4 Position prediction based on historical data

As we are dealing with changing signal environment, existing topology of the building and interference issues - estimation results varies in some degree, depending on the impact of these factors. Sometimes system is not sure which position is the best for user as estimation outcome might be very similar. Our goal is to provide additional information to the system which can help to face indecision and overweight the true position estimation.

4.7.4.1 Motivation

After few cycles of experiments we have observed interesting feature about distances computed between online signals result set and fingerprints in the database (Do not mix it with metrical distances between physical positions!). There are lots of cases when the best candidates' distances (the lowest) differs by a very small amount. Very close similarity we refer to the closest reference points where ratio between the lowest distance and such estimated distance is higher than 0.9. Intuitively considering indoor environment we are running our tests in, we claim that selecting best fingerprint which has slightly smaller or almost equal distance comparing to neighboring fingerprint is not secure decision. Signals lean to fluctuate. Every small change in signal strength makes influence on position estimation. Even standing idle can make system confused about which location you are currently on.

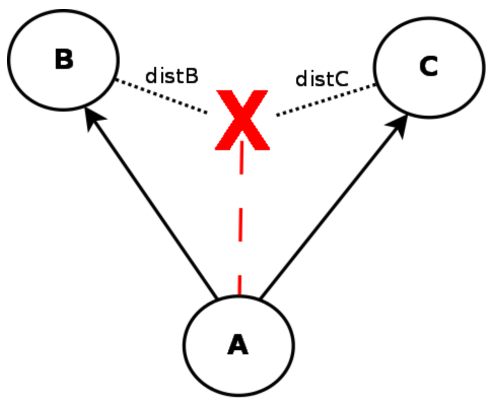


Figure 4.28: Situation when user is in uncertainty zone

In the figure 4.28 situation of uncertain position is presented. Let say your last visited position was A and you have ended up in position X. X is uncertainty zone, where bounds of neighboring fingerprints intersects. In this position system has scanned signals from visible access points. After positioning actions system generated outcome, where $\Delta(distB, distC) = \epsilon$. Selecting $distB$ having $distB > distC$, where $|distB – distC| <= \epsilon$ does not guarantee the estimation correctness.

There is a fact that users tend to walk the same paths. Every day person may go to a canteen or to his office leaving repetitive locations. Relying on this information we can improve accuracy of estimation. We are trying to make an impact on the final outcome application performs. In experiments we try to help system to deal with indecision situation like in figure 4.28 and to underweight distance of reference point which he visits most of the time. We present function and parameters required to determine most visited position. However we leave some of the parameter calibration for the future work. Default parameters are chosen.

4.7.4.2 Analysis

In order to know the past paths of the moving user we have created *poshistory* table. We store historical data as a sequence of positions user have visited. Every set of position, user and time determines one record in the *poshistory* table. Joining these records together by incrementing time the paths can be constructed for every user separately.

We have introduced prediction probability ratio to be able to control the value of estimated distances between online signal result set and fingerprints. Prediction probability ratio denotes the probability of certain position being next after visiting current position. Let say in figure 4.28 there were three times user went to position C and two times was located in B after visiting A. Prediction probability ratio for B would be $\frac{2}{5}$ and $\frac{3}{5}$ for C having A as a current position. This probability is expressed by formula 4.1:

$$moddist = \frac{dist}{(ratio + 1)},\ ratio \in [0, 1] \tag{4.1}$$

We can read this formula as follows: in case where certain fingerprint was used with higher prediction probability ratio distance is maximized, while in other cases ratio will have lowering impact on the distance. This formula is applied only for fingerprints which are very close to online signal vector - their distances to single online signal vector is very similar. However there are situations when ratio can be equal to 0 as some of reference points were never visited before. Therefore to avoid illegal division by 0, distance is divided by sum of ratio and 1.

Proximity of fingerprints is determined by similar distances, where similar distances are defined by distance ratio threshold. Distance ratio identify ratio between shortest distance and distance where certain fingerprint is used. This distance ratio threshold is set to 0.9 in our system. If distance ratio is higher than this threshold fingerprints are considered as close and function is applied for this distances. Calibration of this threshold is left for future work.

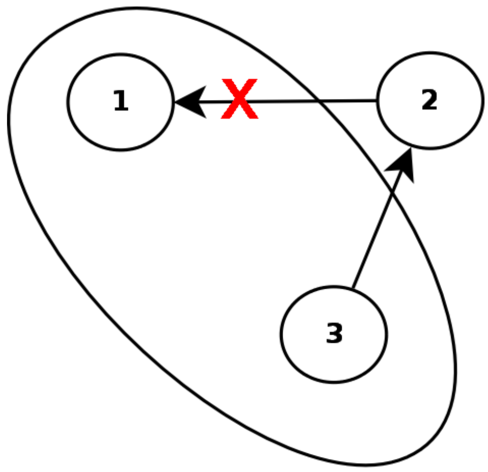


Figure 4.29: Importance of historical positions

After historical data analysis we have noticed that based on recorded data we can say from what direction user was coming and in which direction he was going expressed by position. Path patterns can be constructed. This information can point prediction probability ratio to choose more realistic values. To make it clear, let consider example presented in figure 4.29. Currently user is located in position marked with X and is going toward position 1. Previously user visited position 2 and 3 even earlier. Let name position 3 as history position. Going further, system estimates position and finds two fingerprints from position 1 and 3 close to online signal vector (fingerprints' proximity). Suppose, we have following history from *poshistory* table:

Trajectory	Frequency
2->3	5
2->1	3

As the result distance from fingerprint 1 will be underweighted more than fingerprint 3. And predicting in this case will worsen outcome. However if we consider past positions as a basis for knowing from where user had come, the result may not be worsen.

Trajectory	Frequency
1->2->3	5
3->2->1	3

After using historical node 3 distance will be lowered between fingerprint 1 and online signal vector. No impact will be made on fingerprint 3 as there are no such paths where after position 3 through position 2 it reaches position 3. As we claimed, user tends to walk the same paths. This assures position to be estimated in as a most probable location. Of course intuitively we can ask what if user changed his mind and turned back. The answer is that there are not many such situations which can distort results. Moreover here is good explanation why we use only 0.9 distance ratio threshold. This threshold actually can limit such situations as presented in the picture 4.29 , where metrical distance between fingerprints is reasonable high.

It is worth to mention the fact, that granularity of our radio map is coarse, where density is only 4 meters in average. Thus as we make scanning every second, usually same position is recorded few times until it reaches next position: (A, t_1), (A, t_2), (A, t_3), (B, t_4), (B, t_5)... These kind of repeated loops are ignored in our algorithm. As we assume, that they are providing not much of the useful information. Only unique positions are taken into account. Moreover if we take these loops into account prediction probability ratio will be changed. If we consider example 4.28 and similar fingerprints A and B having position A as a starting point, ratio in most cases will overweight for A, as self-repetitions are recorded more often than transitions. Transition we call a switch from position X to position Y, where position X is not the same position as position Y. However in the future work we can exploit loops in more details. After mining self-repetitions from *poshistory* table we may more precisely define upcoming position user is about to visit.

4.7.4.3 Algorithm

SQL function 3 *predPosRatio* was developed in order to compute prediction probability ratio for every case where difference of fingerprint's distances to online signal vector is not clear. We employ few parameters: *last_pos_id* – last position identification, *user* – user identification, *curdate* – current date, *mindelta*– amount of minutes where positions are still considered as a trajectory and *hcount* denotes amount of historical positions used in position prediction. We use *mindelta* as there is situation when user logs his location before one day – we do not treat these positions whose time difference is higher than *mindelta* as connected. Finally function returns prediction probability ratios with all computed transitions. Function results are only used with close fingerprints. For simplicity reasons and lack of the time only default parameters were used in experimental part. *mindelta* = 10 min, *hcount* = 1. Parameter variation is left for future work. Our task was to examine if this kind of history based prediction probability ratio can make positive influence on the accuracy of the system.

Function `predPosRatio`(*cur_pos_id, user,curdate,mindelta,hcount*)

Output: *next_transitions*

1 *next_transitions* $\leftarrow \emptyset$;
2 $x_0 \leftarrow$ *cur_pos_id*;
3 *hist* $\leftarrow \{ x_1,...,x_{hcount} \mid x_i \neq x_{i+1}$, time($x_i$) - time($x_{i+1}$) $<$ *mindelta*, i $\in \{ 0,1,...,$*hcount*-1 $\}\}$;
4 **if** *hist* $\neq \emptyset$ **then**
5   **for** $\forall$ *position* $\in$ *history(user)* **do**
6   **if** *position* = *cur_pos_id* **then**
7       $x_k \leftarrow$ *position*;
8   *newhist* $\leftarrow \{ x_1,...,x_{hcount} \mid x_i \neq x_{i+1}$, time($x_i$) - time($x_{i+1}$) $<$ *mindelta*, i $\in \{ 0,1,...,$*hcount*-1 $\}\}$;
9   **if** *newhist* = *hist* **then**
10   **if** $x_{k-1} \notin$ *next_transitions* **then**
11   **append** x_{k-1} **to** *next_transitions*;
12   **end**
13   **else**
14   **increment** counter of $x_{k-1} \in$ *next_transitions*;
15   **end**
16   **end**
17   **end**
18   **end**
19   **return** *next_ transitions*
20 **end**

In a first stage of the algorithm 3 (line 3) ordered set of past *hcount* positions connected with last position is selected. x_i denotes record in the *poshistory* table where i varies between 1 and amount of records in *poshistory* table. Incrementing *i* index means older record in *poshistory* table. As it was mentioned before we apply time condition between positions (only unique neighboring positions are considered as it was stated earlier 4.7.4.2). Time function in algorithm is defined as time(*position*) = {position} $\rightarrow$ {time when this position was visited} . If time between at least two unique neighbor positions is higher than *mindelta* ordered set of past positions is considered as empty set. And if there is no history available (line 4-20), empty set of ratios is returned by function. In algorithm: history(*user*) = {user} $\rightarrow$ {set of historical positions}. Going further for every position in history belonging to the user (line 5-18) function searches for position which is equal to the last position. If such position is detected (line 6-17), it's historical pattern is being examined. *newhist* denotes ordered set of past *hcount* positions starting from detected position x_k. If same pattern is discovered (line 9-16) as from the *last_pos_id* – function gets next transition starting from x_k position. Next transitions and their frequency (line 10-15) are collected and prepared as a return variable. *next_transitions* is defined as set of {next transition, amount of such transitions}. As it was mentioned earlier in the ordered set of historical positions no duplicates exists as we do not consider repetitive positions.

There is also additional parameter not mentioned in this function, as it was not used. This parameter allows computing position ratios only for reachable position from the last position. In other words next transition can be made only if last visited node is connected by edge with predicted node. This option allows filtering such fingerprints which position is not connected with the last position. Reachability value can be varied as in the case of graph experiment. Due to the limited time this possibility was not exploited in this function.

4.7.4.4 Results

In this section results of experiments are demonstrated.

We have run this experiment in simulation mode in order to test the impact of prediction probability ratios on position estimation accuracy. 50 random routes are chosen by application to test the case. Before simulation start *poshistory* table was not populated with data. User's history was generated dynamically during experiment. It means that only after visiting particular route at least once - prediction ratio function was able to calculate prediction ratios. Method was fully utilized only after few rounds when particular route was employed.

We have performed experiment twice with different test signals data as input. Once it was taken from Realtek RTL8187L NIC, next time – HP WLAN W400. As training signals we used signals collected only with Realtek RTL8187L NIC.

Before result description some meanings need to be elaborated. In the table 4.5 effect of prediction is explained. In deterioration prediction changed position not into true one. Improvement means that prediction ratio has good influence and estimated position was changed to true one. There is others effect when prediction changed position but neither changed position nor estimated was correct. Impact denotes all cases when outcome position was changed to different position than estimated. No impact occurs when position changed by prediction is the same as estimated.

Effect of prediction	Estimated	True	Changed by prediction
Deterioration	1	1	not 1
Improvement	2	1	1
Others	1	2	not 1 and not 2
Impact	1	2	not 1
No Impact	1	x	1

Table 4.5: Example of prediction's effects on position estimation

In the first experiment we got results as shown in the figure 4.30 where Realtek RTL8187L NIC input data was used. Every specific effect is expressed in percentage of all estimation made in particular position. Diagram was minimized as in positions not included in the graphic prediction has no impact on position estimation. In all three positions and every time when prediction was used it brought improvement in estimation.

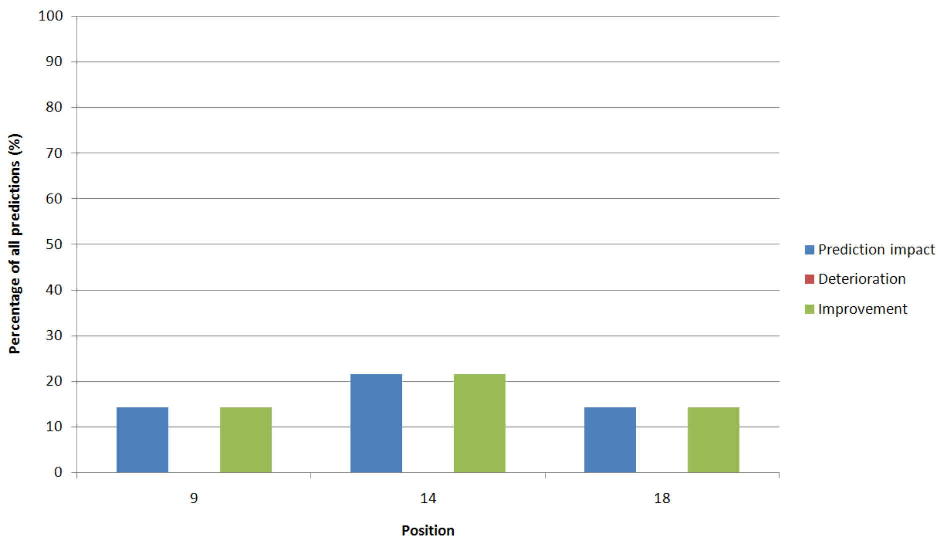


Figure 4.30: Result of prediction effect with Realtek RTL8187L

In the second experiment results are totally different. Comparing to the previous experiment input test signals are taken from HP WLAN W400. In this case almost in every position prediction had an impact on estimation. In six positions 1, 3, 12, 13, 14, 15 deterioration reaches 100% of the predictions which have impact on position calculation. Only in three positions we can notice improvement (2, 4, 9). There are 8 positions (6, 7, 8, 10, 11, 16, 21, 22) where prediction had changed estimated position, but it did not worsen or improved (others situation mentioned in the table 4.5) results. Maximum improvement (35.7% of all estimations at that position) is detected in position 2. The worst case is detected in position 1 with highest deterioration level (85%).

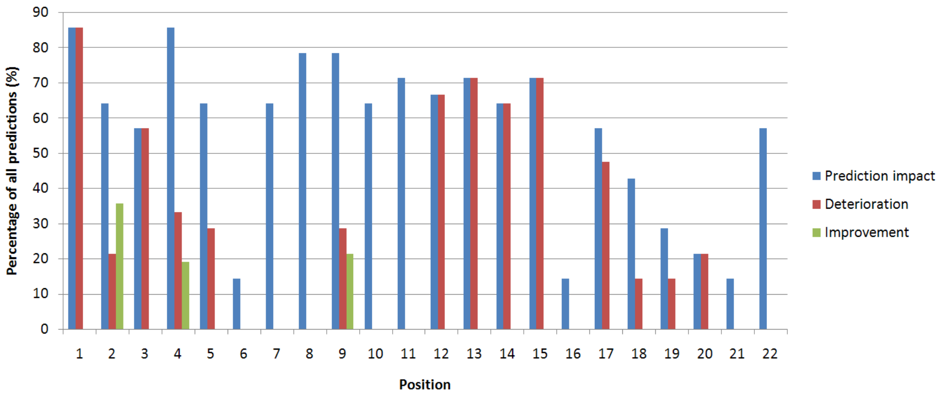


Figure 4.31: Result of prediction effect with HP W400

To sum up, in average 29% of all estimation cases prediction had deterioration impact and only 4% brought improvement. In this case we lost more accuracy of position estimation instead of getting better.

4.7.4.5 Conclusions

We have performed two experiments in order to show that some users can generate incorrect history and based on that incorrect history prediction technique can even worsen accuracy of position estimation. Second experiment showed that having device with good specifications and additionally having data of offline phase signals available in database, can lead to improvement of position estimation.

After experiment we have measured distance between collected online signal strengths by HP WLAN W400 in particular position and available fingerprint recorded by Realtek RTL8187L NIC. Value of this distance was different enough to claim that HP WLAN W400 has worse signal perception quality than Realtek RTL8187L NIC.

This feature must be employed with exceptional caution depending on category of users' device.

As mentioned in earlier sections this technique opens lots of space for examining different parameters and situations. The impact of prediction probability ratio, proximity threshold of fingerprints and size of historical positions may be varied depending on different NICs. Consider repetitive positions in prediction probability ratio. There is also possibility to incorporate historical trajectories collected from different users with lower influence.

Be aware that in real life case results may be more inaccurate.

4.7.5 Enabling directionality

Being able to predict in which direction user will be moving can improve location determination and also facilitate the process of navigation in the indoor space. There are huge buildings such as clinics and airports where value of navigation is very high. Direction identification can also be helpful for blind people as it can serve as a feasible direction guide.

In this section feature of direction determination is presented. We try to investigate whether it is possible to determine direction, based on the orientation in which fingerprints were recorded. Graph model and history of user trajectory was being also employed in order to be able to predict future direction of the moving object.

Results and conclusions of experiments are also demonstrated in this section.

4.7.5.1 Offline phase with orientation

Inspired by [15] and [29] we have collected directional fingerprints. To make it more precise our goal was to examine how much fingerprint is influenced by orientation of mobile user. Orientation here means in which direction user's antenna is pointing. Like in the paper [15] four cardinal directions were considered: North, South, East and West. This refers to user facing one of these directions while fingerprints were recorded.

After collecting signals in one position with different orientation analysis was made in order to check how different signals are, considering user's direction. We computed averages of all signal strengths belonging to specific directional fingerprint in order to compare directional fingerprints. In figure 4.32 average SS of four directional fingerprints is presented. Difference between fingerprint's average SS fluctuates from 1 up to 4.6 dBm. Between opposite direction (N-S, E-W) difference is 4.5 and 1.4 dBm respectively. Moreover we have noticed that distinction between neighboring direction averages is less significant ranging from 0.8 dBm to 2.5 dBm. All in all as we can see in figure 4.32 difference is not very high.

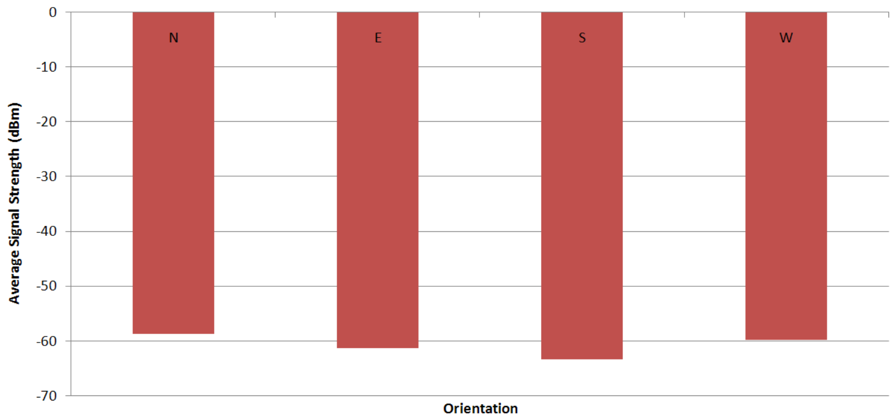


Figure 4.32: Average SS fluctuation in directional fingerprint

In order to increase the diversity between directional fingerprints we have introduced the offset value. Offset is the distance from the center of the reference point oriented to the one of directions with specific value. We have tried only small value equal to around 0.3-0.5 meters. Only small offset was considered, as making distance larger can negatively influence position estimation: neighboring fingerprints may become more similar. Variation of the offset value is left for future work. However applying offset results have showed almost no changes. Fluctuations between directional SS averages have almost not changed (0.8 dBm up to 3.87 dBm). Difference between opposite direction fingerprints was 2.1 dBm in case of North and South and 2.7 dBm in case of East and West.

To test efficiency of fingerprint's orientation 30 scans with each direction was collected in one position (number 6 on the map 4.8). After that we scanned signals (100 times) in specific direction and using Euclidean distance formula, distance between signal vector and four directional fingerprints was computed. The outcome was the direction with the smallest computed distance – which means that particular fingerprint was the closest to directional signal vector. In figure 4.33 results of this experiment are presented. In this case accuracy in the figure means percentage of correctly determined direction. As we can notice this accuracy in all cases does not reach 60%. The highest accuracy of 53.3% is obtained when user orientation was north, while in case of the south – lowest accuracy of 33.3%. All in all, the level of correctly determined direction is not very high.

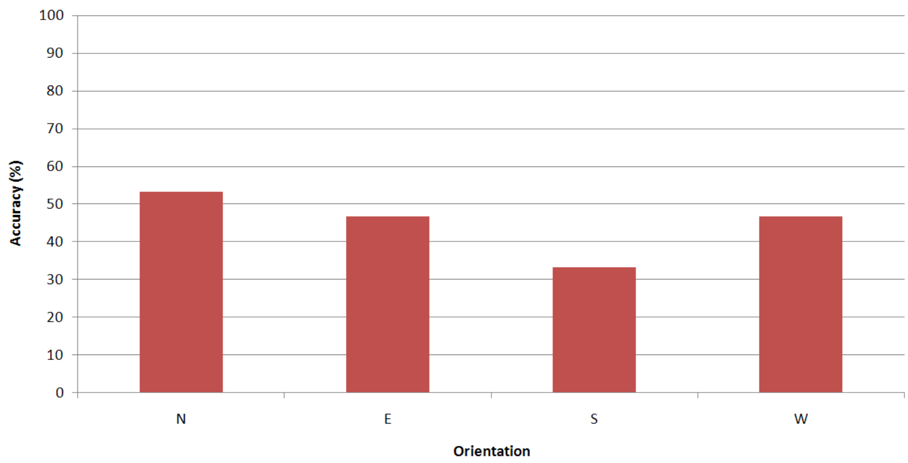


Figure 4.33: Direction accuracy based on directional fingerprint (4 dir.)

Based on evidence from figure 4.32 where fingerprints collected in opposite direction were more different than neighboring, we tried to determine direction utilizing only two opposite directional fingerprints. In figure 4.34 results of this experiment are demonstrated. As we can see accuracy increased up to 80%. 75% was recorded while testing west orientation.

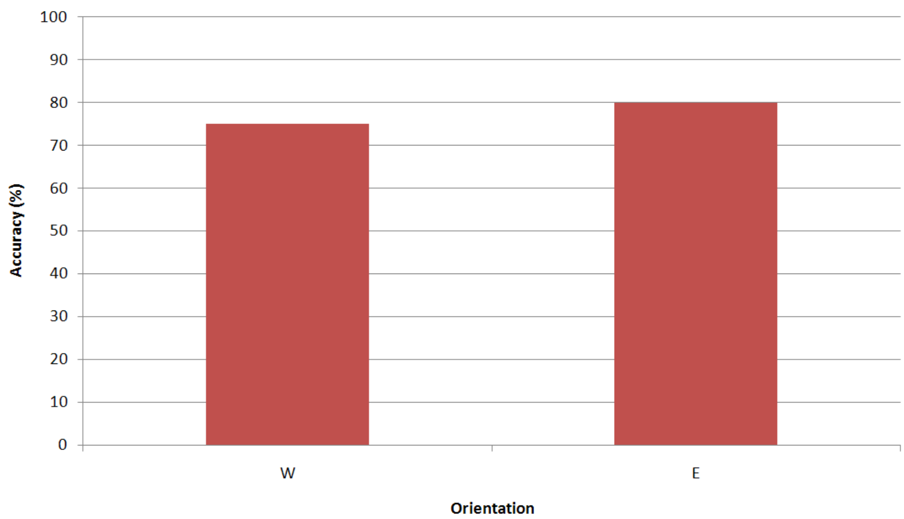


Figure 4.34: Direction accuracy based on directional fingerprint (2 dir.)

To sum up, these results showed that if more orientations are considered, lower accuracy can be reached as it is difficult to make distinction between them. Later this feature is used to employ only possible orientations. For example, in the corridors there are only two directions user can possibly move.

We also took in mind the fact that it was tested only in one reference point and in the static conditions. Populating radio map and considering user motion, accuracy will decrease due to signal more chaotic reflections and proximity between fingerprints in the radio map.

4.7.5.2 Direction determination

In this subsection we present the method how we determine future direction of the mobile user.

As it was reported in previous subsection relying only on directional fingerprints can bring not very stable and accurate results. To determine user's direction we try to employ historical and physical constrains. Graph model adapted to the radio map reference points can give information to the system about connected and possible reference points or nodes from the current position and in that way limit the possibility of direction range. On the other hand based on history we can say in which direction user tends to walk.

Before explaining how direction is actually calculated it is important to mention about the way we determine the direction having two positions. Firstly in the figure 4.35 we can see four cardinal directions associated with our map. Every cardinal direction belongs to specific range of degrees: South $\in$ [-45, 45), East $\in$ [45, 135), North $\in$ [135, 180] $\cup$ (-180, -135), West $\in$ [-135, -45). In order to calculate the direction we utilize arc tangent function and compute the angle between axis N-S and line which goes through the specified coordinates (pixels) of two positions. For example vector $\overrightarrow{4,25}$ denotes direction West, while $\overrightarrow{7,19}$ – South.

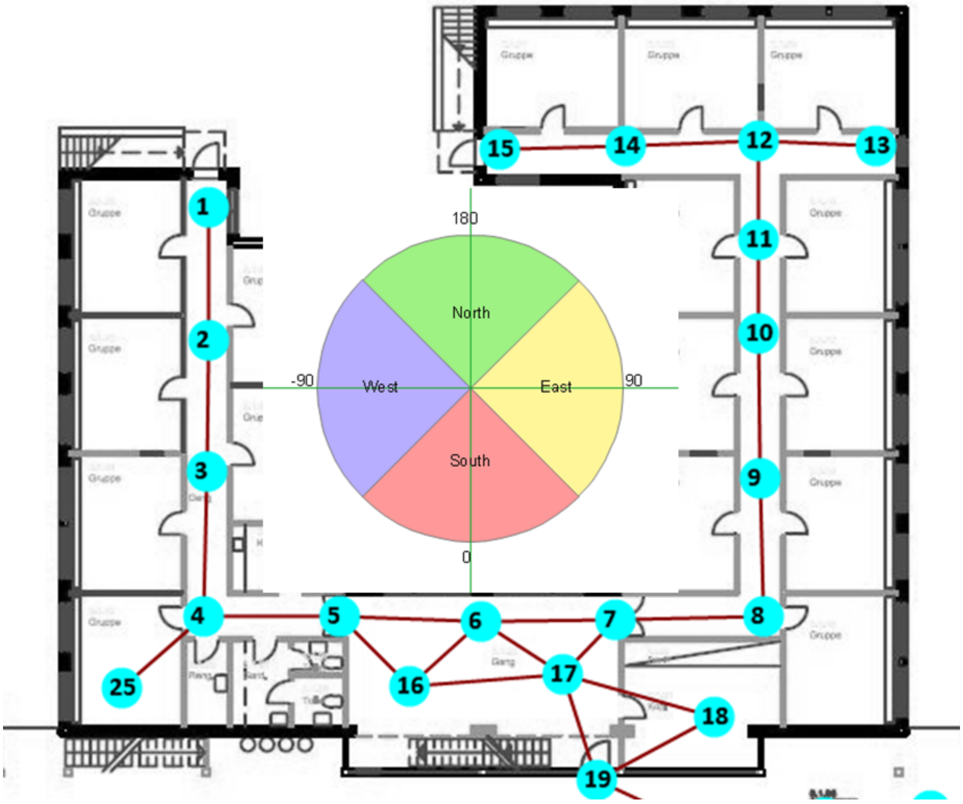


Figure 4.35: Cardinal directions on the map

In our direction determination method three cases in which mobile user can be are distinguished:

1. one possible position exists (end of the graph)
2. two possible positions exist (usually corridors)
3. more than two possible positions exist (intersections)

For example in position 25 possible node is 4, in position 2 – node 1 and node 3, in position 4 – node 25, 3 and 5. As you can see graph model is employed in order to define such situations.

For every case we have applied different computations.

In the first case when user has only one position to go direction is estimated trigonometrically between current position and possible position. For example in position 15, direction will be always east as vector $\overrightarrow{15, 14}$ belongs to [45, 135) degree range.

In the second two we check last position user has visited. This position is ignored as a future possible position for simplicity reasons and due to the fact that users do not tend to change their direction very often. After last position elimination we apply trigonometric formula and estimate the region where cardinal direction resides. For example if user came from position 1 and currently is in position two estimated direction will be south.

Third case is more complicated. In this situation we try to exploit and combine values of prediction probability ratio 3 and relation of current signal vector and directional fingerprint in order to decide the most probable direction. In this case few steps are considered to select best suitable direction:

1. Define possible positions based on prediction probability ratio
2. Find possible cardinal directions based on possible positions
3. Compute proximity between signal vector and directional fingerprint (possible directions only)
4. Combine prediction probability ratio and fingerprint proximity ratio
5. Select direction with maximum probability

Prediction probability ratio is computed using 3 procedure. There is only one historical position considered in this formula. Let say current position is 4 and historical is 3 (check the figure 4.35). From history we get information that with such trajectory pattern user went to position 25 twice and to position 5 even 25 times. Based on available ratios we get two possible directions: west and east, for west we have ratio $\frac{2}{27}$ and east $\frac{25}{27}$. North direction is ignored as prediction probability ratio does not exist for position 3. In the third step as we already have possible directions we compute Euclidean distance between fingerprint belonging to possible direction and signal vector. Minimum distance is recorded. In the fourth step we have introduced formula to combine two values 4.2:

$$P(dir) = \frac{4}{5}\,predRatio(dir) + \frac{1}{5}\,proxRatio(dir),\ dir \in possible \tag{4.2}$$

Proximity fingerprint ratio is computed as ratio between distances: minimal distance and the current distance.

$$proxRatio(dir) = \frac{mindist}{dist(dir)}$$

For example if we have estimated distance showing proximity with "east fingerprint" equal to 100 and "west fingerprint" equal to 120, then proximity ratio for east will be 1, and for west – $\frac{5}{6}$. Larger distance means lower ratio and less influence on choosing particular direction. Continuing example probability for east will be equal to 0.91 and for west - 0.26. We have chosen constants $\frac{4}{5}$ and $\frac{1}{5}$ in order to lower impact of proximity ratio and increase the impact of history. As we have seen earlier 4.32, average of signal strengths tend to vary between directional fingerprints in a very small degree. Because of that we cannot expect very high direction accuracy thus we cannot rely on this equally as history. While historical data gives tendency of user movement which brings more stable outcome. Due to the time limitation we did not consider other variations of constants.

4.7.5.3 Results

In this experiment we have run simulation to test direction prediction. Additional features (graph, regions) of the system were ignored in order to make results more clear. Moreover before test there was no history generated for user. We used test signals collected by Realtek RTL8187L. Direction prediction is considered as correct if direction between true positions is the same as direction estimated previously by formula 4.2. Accuracy denotes the level of correct predictions.

In the figure4.36 we show results we got after the test. We have divided simulation results into few sections. In every section growing amount of predictions or visited positions is presented. After 50 direction predictions (50 visited positions) accuracy reaches only 66% while after 1200 predictions it reaches 93.41%. After 300 positions were visited accuracy is stabilizing. Fluctuation between second case and third is only 3.58%. Direction prediction accuracy grows up proportionally to amount of historical data what is apparently visible in the figure 4.36.

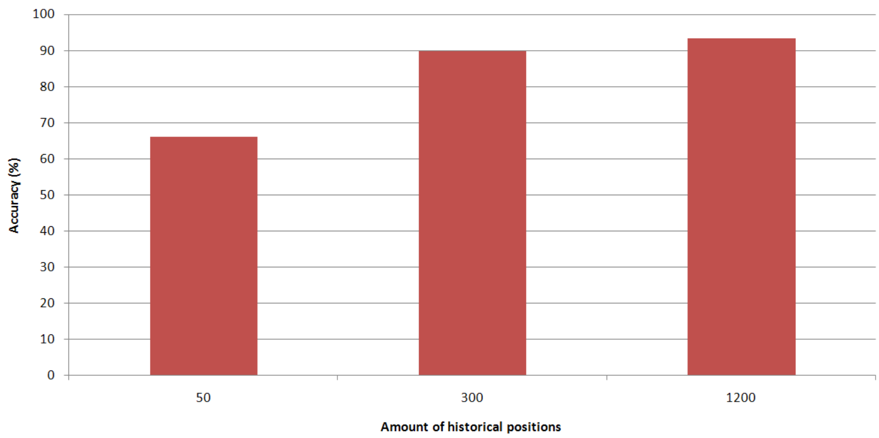


Figure 4.36: Direction accuracy depending on amount of historical data

4.7.5.4 Conclusions

As our solution is based mainly on historical data, more history means better direction prediction. However as it was mentioned in previous section, if accuracy of certain device is very low, recorded history will be less accurate thus it will distort direction prediction results.

Relying only on directional fingerprint information will not bring high direction accuracy unless directional fingerprints are reasonably dissimilar. In our experimental environment where signals strengths of access points tend to fluctuate, small variation between directional fingerprints might be vanished after certain amount of time. If granularity of radio map is coarse then collecting directional fingerprints with higher offset value may improve direction predictability. However to get more precise results it is left for future investigation.

4.7.6 Combining different infrastructures

As mentioned at the beginning of this report - one of the main ideas behind this project was to find a way to combine different available infrastructures into one system and experiment with how can that increase positioning accuracy. We have chosen to use Bluetooth because it is available in most of today's electronic devices and there is no need to invest extra funds on the additional equipment.

We base our main idea on the fact that Bluetooth infrastructure is designed for low power consumption and comes in three predefined standards: Class 1, Class 2 and Class 3 as shown in table 4.6.

Class	Operational Range
Class 1	∼100 meters
Class 2	∼10 meters
Class 3	∼1 meter

Table 4.6: Bluetooth specification table [2]

Because of the relatively low range compared with Wi-Fi, Bluetooth stations can be used to detect clients that come into their scanning range very accurately. It means that if Class 3 device is used and it detected some specific user's Bluetooth-enabled portable device – we can be sure that the user is within 1 or 2 meters away from the actual station.

The advantage of this approach is that the position of the user can be determined with a very high precision depending on the range of the station. However if we were about to cover a large indoor area - it would require a great number of Bluetooth stations which is both not practical and expensive. On the contrary - Wi-Fi infrastructure can cover a relatively large area but at the same time makes it difficult to estimate the exact position of the user. By finding a way to combine advantages of both infrastructures we can noticeably improve the accuracy of indoor positioning system.

4.7.6.1 Dividing indoor space into regions

In this experiment case we use Bluetooth devices in order to divide larger indoor space into smaller regions. Similar to [10] - the idea is to place a Bluetooth device in a strategic location in the indoor space in such way that it could separate the indoor space into two separate areas. It is very important to arrange everything in such way that when user walks from one region into the other – he or she can only do that by being detected by one of the Bluetooth stations. Figure 4.37 illustrates a possible way to divide indoor space into 4 separate regions using Bluetooth devices labeled as 5, 7 and 19.

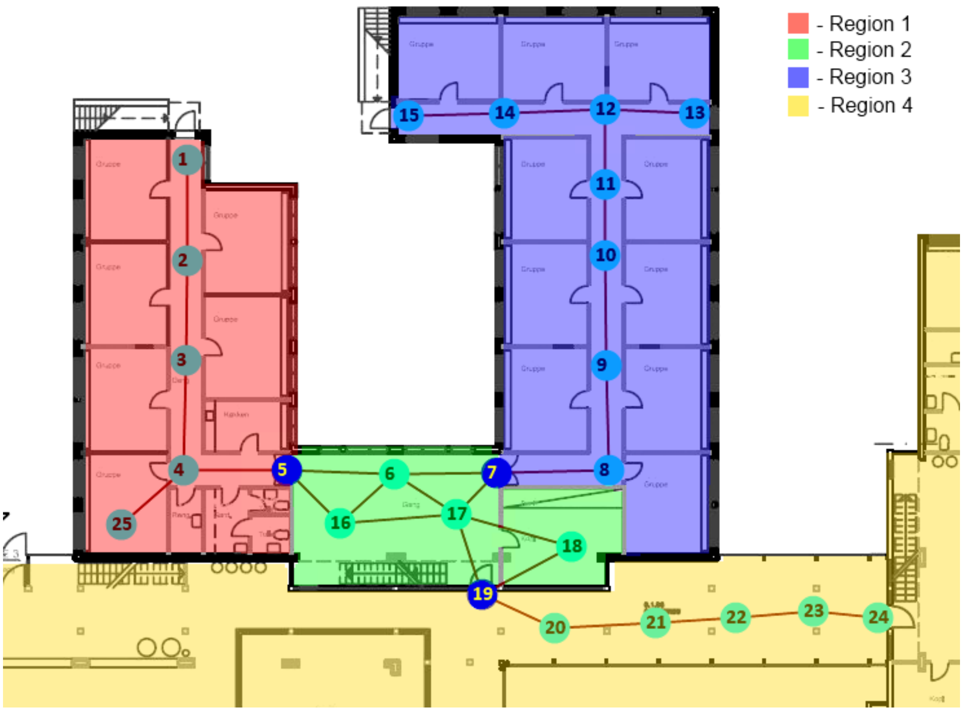


Figure 4.37: Dividing indoor space into regions

As figure 4.37 suggests – it is a good idea to place such Bluetooth hotspots in narrow entry spaces such as corridors or doors which connect two different halls. By doing that - you make sure that the Bluetooth station will be able to detect the user every time he or she enters a different region.

Similar to the graph case which was documented earlier - this enables us to reduce the space of user's possible positions. Once we know that the user has entered a particular region – we are sure that he or she will stay there until we get an indication from a Bluetooth station that the user is entering another region.

By following this logic we have implemented our indoor positioning system in such way that the set of possible positions of the user always belong to one or two regions at a time. When we know that the user is currently in region 1 then the set of all the future possible positions of this user belongs only to this region until some of the Bluetooth stations detect his movement. If indoor positioning system knows what region the user is currently in – during the position estimation phase it only needs to consider those reference points that are within the area of that particular region.

Using information gathered from Bluetooth devices the system defines all the possible positions of the user. Later – using Wi-Fi it tries to estimate a more precise position of the specific user. In short – first Bluetooth infrastructure is used to determine what regions the user might be in and later Wi-Fi is used to predict a more precise position within that region. This technique guarantees that the maximum average error distance is always smaller or equal to the size of the region.

There are some specific cases where system cannot tell exactly what region the user might be in. Using the example from figure 4.37 imagine a case where user being in region 1 comes close to Bluetooth station labeled 5. In that particular time the system cannot always guess what region user is going to enter. The most probable candidate is region 2 but is not always the case because user can always turn around and get back to region 1. To solve this problem, we make sure that whenever user is standing in the Bluetooth hotspot then the set of all possible future positions is composed from those regions that the Bluetooth device is separating. In the previous example the set of all the possible positions of the user would be composed of reference points from region 1 and 2. Only when the system is sure that the user has entered region 2 and receives no indication from any of the Bluetooth devices – it then makes sure that only reference points that belong to region 2 are going to be considered while estimating the position.

We have implemented our system the following way – whenever we receive an indication that the user is within detection range of one of Bluetooth devices – we instantly report that position as user's current position. This is because detection range of our Bluetooth devices was reduced down to 1 or 2 meters and if we can determine a position of the user with that accuracy – we do not need to do any further estimations.

When user is detected at one of the Bluetooth hotspots – system always return that position as users most probable position. Later the system tries to figure out what region the user might most probably be in. In most cases each Bluetooth device connects two regions so later it is important to determine where the user went upon leaving the Bluetooth hotspot. After the client is no more detected by the Bluetooth device – the system does 5 scans using the wireless network adapter (it takes at least 5 seconds in our case) in order to detect what is the current region of the user. After those 5 scans it evaluates the results and checks what region out of those two is the most feasible. We have chosen number 5 because of the fact that the results cannot be divided equally among two regions. After determining the current region of the user – system then reduces the possible future locations of the user down to the reference points that belong the current region. This approach guarantees lower average error distance as well as it improves the performance of the system.

4.7.6.2 Additional techniques

Another way that we try to utilize this infrastructure is to make our system self-maintainable to the highest degree possible requiring no user interaction if possible. As described above – Bluetooth can give us the precise position of the user. By knowing this ground truth – we can automatically maintain our radio-map. This is achieved in the following way – whenever the system detects that the user has entered a Bluetooth hotspot – it then automatically scans for available access point signal strengths and uploads the results to the database together with the position that indicated by the Bluetooth device. That way we can force all the users to update our radio-map on a constant basis without having them to do anything.

A number of other useful techniques can be implemented by having such low range infrastructure combined with Wi-Fi. As one of the features of our system, we have assigned each Bluetooth device a speciffic id. Each such id of device that is positioned on ground floor starts with a digit 1 and increases according to what floor the device installed on. This is useful because when the system detects that the user is within range of some particular device – it then checks what floor this device is installed on and if it notices that the user is using the plan (map) of another floor – automatically changes it to match the floor that the user is currently in. In order to get the most from this feature it is advised to have a Bluetooth hotspots positioned near the stairways at each floor. Then the map will automatically be changed to match the floor with the minimal delay.

It is usually a good idea to deploy such Bluetooth devices in the area where Wi-Fi coverage is fairly low. That is dictated by the fact that low strength Wi-Fi signal is less stable and it makes it very hard to estimate the correct position of the user.

4.7.6.3 Results

In this section we present results from two different experiments while using Bluetooth devices to divide our indoor space into different number of separate regions. Later we compare this approach with the initial case where no Bluetooth devices were used.

Accuracy comparison

In order to see the advantages and disadvantages of this approach we have first made an initial test in order to record what is the current accuracy and average error distance of the system without using any of the Bluetooth devices.

Later, we divided the indoor area into 4 separate regions placing Bluetooth devices on the strategic narrow places.

Figure 4.38 depicts the environment and placement of the Bluetooth devices labeled as 5, 7 and 19.

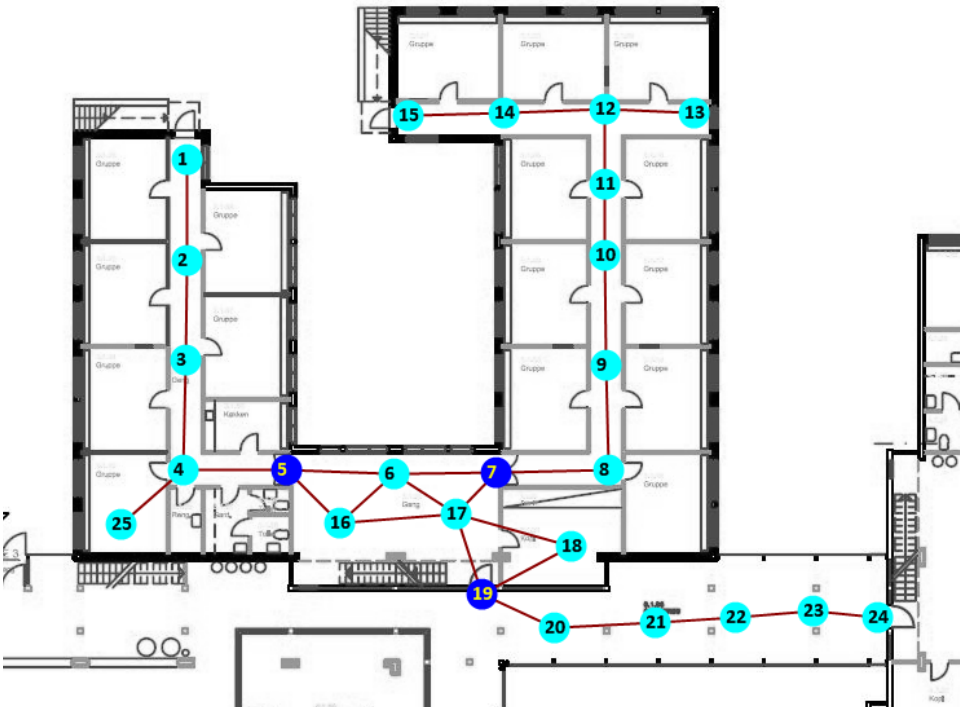


Figure 4.38: Dividing indoor space into 4 regions

Having recorded all the accuracy and average error distance results we have then introduced additional Bluetooth devices. By conducting this test we wanted to observe how the number and size of the regions would affect the overall performance of our system. In this case the indoor space was composed of 6 smaller regions as illustrated in figure 4.39.

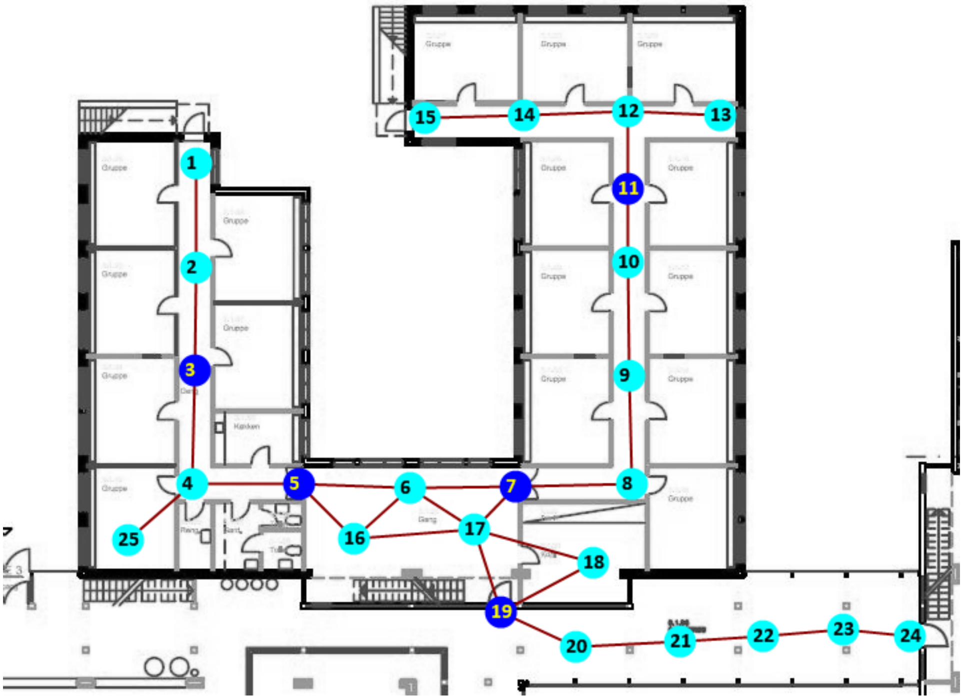


Figure 4.39: Dividing indoor space into 6 regions

Figure 4.40 presents a summary of system's average accuracy in each position while using three different approaches described above.

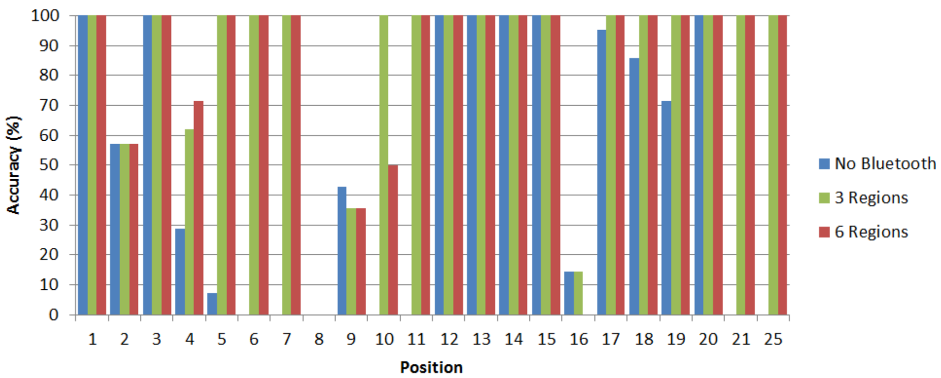


Figure 4.40: Accuracy comparison

It is obvious that there was a noticeable improvement in systems accuracy by utilizing the Bluetooth infrastructure. At 11 out of 25 positions there was a noticeable increase in accuracy. The most significant improvement was achieved in positions 5, 6, 7, 10, 11, 21, 25 where compared with the initial case system's accuracy has increased up to 100%.

There was only a single position labeled 9, where accuracy has dropped when introducing Bluetooth.

Average error distance

This section presents average error distance measures of each case.

Figure 4.41 shows the average error distance of the initial case where no Bluetooth devices were used.

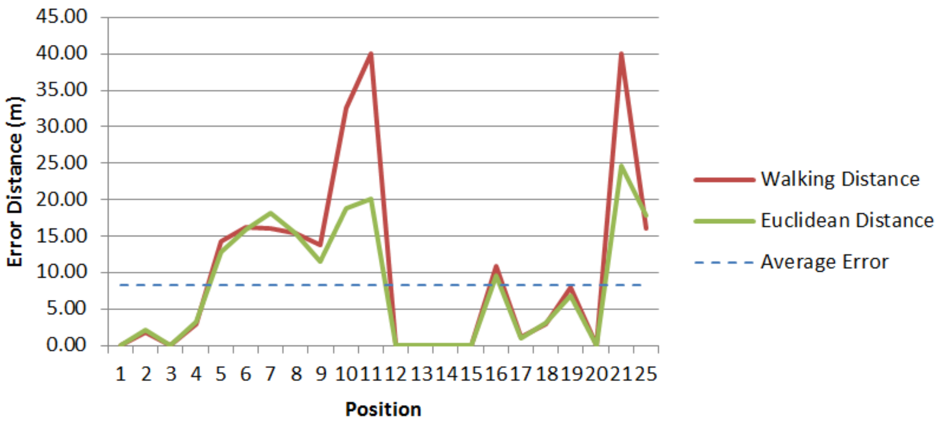


Figure 4.41: Average error distance without regions

As it can be seen – Euclidean error distance at positions 10, 11 and 21 reaches 19, 20 and 25 meters correspondingly. The average error distance among all the positions remains at 8.24 meters.

If we examine the walking distance error in each position - we can clearly see that there are two major peaks in the diagram. At positions 11 and 21 it reaches a 40 meter mark which clearly indicates that this approach is surely not acceptable to be deployed in such an environment that we are currently dealing with.

The following figure 4.42 illustrates the average error distance of a test where indoor space was divided into 4 regions with the help of Bluetooth infrastructure.

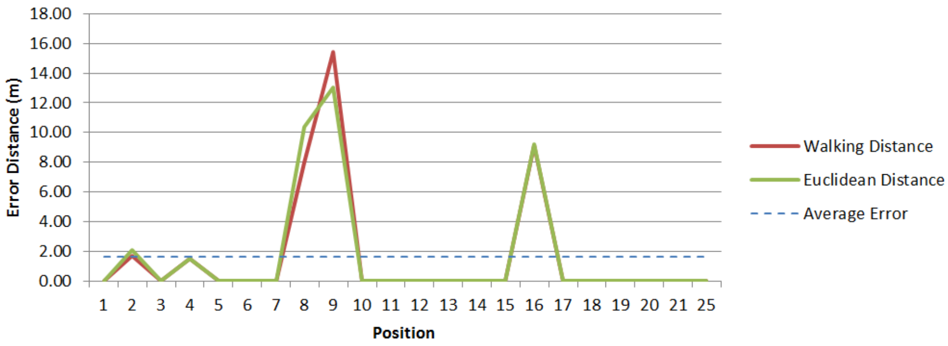


Figure 4.42: Average error distance with 4 regions

This time the error distance stays minimal in all the positions except in those labeled as 8, 9, and 16. The absolute maximum Euclidean distance error reaches 13 meters in position 9, while maximum average walking distance error is a bit higher – 15.4 meters, recorded in the same position. In all the other positions the error distance measure is very low. This is the reason why average Euclidean distance of all the positions is only 1.64 meters.

A case where indoor space was divided into 6 regions produced very similar results to the previous test. As figure 4.43 indicates – the results are almost identical and follow the same trend in all the positions. The average Euclidean error distance is slightly higher – 1.87 meters.

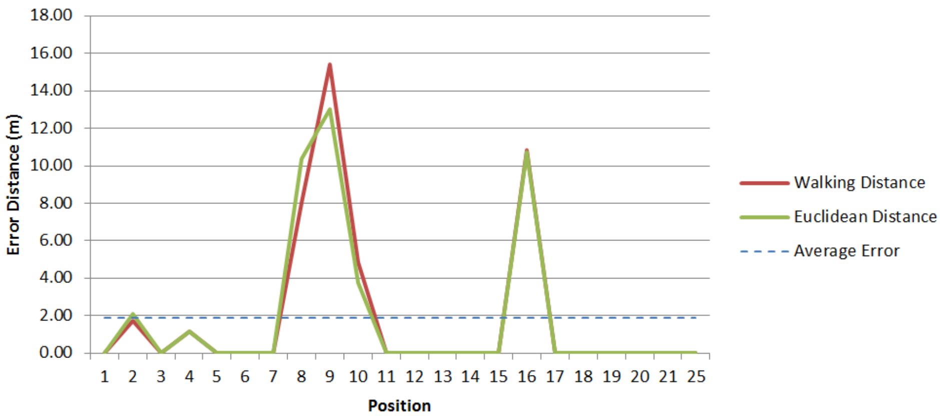


Figure 4.43: Average error distance with 6 regions

4.7.6.4 Conclusions

In general – by using this approach we were able to achieve a significant improvement in terms of accuracy and greatly reduce the average error distance. Another speculation that can be derived from the results of this experiment part is that increasing the number of regions and at the same time making them very small – does not guarantee to bring further improvement. As the last test showed – it is not worth the effort to divide the indoor space in regions that contain only a few reference points.

Compared with the initial case – overall accuracy has improved by approximately 35%. From 50% in the initial case it increased to up to 82% and 85%in the latter two cases.

The average Euclidean error distance dropped from 8.24 meters to 1.64 and 1.87 meters. That is an improvement of approximately 80%.

To conclude – it is important to mention that it is worth the time to experiment with the optimal size of each region. As results of this experiment have shown – increasing the number of regions does not guarantee that it will produce better results.

4.7.7 Real life experiments

In this part of our experiment section we will present the results that were achieved while testing the system in a real world scenario. No simulation was used this time.

In order to test system's performance we need to know the exact position of the user and later compare it with the predicted one. In order to get this ground truth we have implemented a special "Experiment" mode in our system as described earlier in the user interface section 3.7. Whenever we switch to that mode – we are enabled to indicate our current position by pressing anywhere on the map. The system then tries to predict our position and inserts the result together with the actual current position that we most recently indicated into the database. Using this way the user is responsible of reporting the correct current position accurately. Otherwise – the test will provide misleading results. The other question that arises when using this evaluation approach is:

When should the user indicate that his position has changed from position A to position B?

In other words – when should the user click on position B on the map to indicate that he is currently stationed there?

We answer this question based on logic behind Euclidean distance. It should be done whenever we cross the half way between position A and B as shown in figure 4.44below.

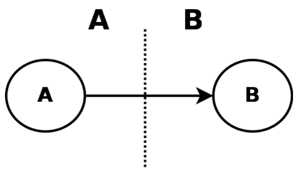


Figure 4.44: Separating positions

The area to the left side of the midpoint belongs to position A, while the area to the right belongs to position B.

We were trying to follow this method carefully while testing the system in this real world case. The experiments of this case were conducted using Asus EeePC. Each experiment took about 10 minutes. We were walking in a quite slow pace and were constantly indicating our current position to the system so that it can later compare it with the predicted one.

4.7.7.1 Wi-Fi and Bluetooth coexistence

In the first part of this experiment case we were using both Wi-Fi and Bluetooth infrastructures. Because we were not able to get Bluetooth stations – we were forced to use our laptop to scan for discoverable Bluetooth devices – phones in our case. It means that there were two simultaneous scans running in the background. We scanned for available Wi-Fi signal strengths once every second, and every three seconds performed a Bluetooth scan. As mentioned before those time periods were dictated by our current hardware and API.

Before this test we have decided to experiment and investigate on how significant is the level interference between the Wi-Fi and Bluetooth wireless signals.

In this section we present a summary of how Wi-Fi scanning results might change in different devices while performing Bluetooth scans simultaneously.

To measure it – we used two different class devices – Asus EeePC and Compaq nw8440.

In the first test we were trying to measure how the average signal strength would differ if we had Bluetooth scanning enabled. We have done 35 scans while having Bluetooth disabled and later we repeated the same procedure with having Bluetooth enabled. Figure4.45 presents the results of Asus EeePC.

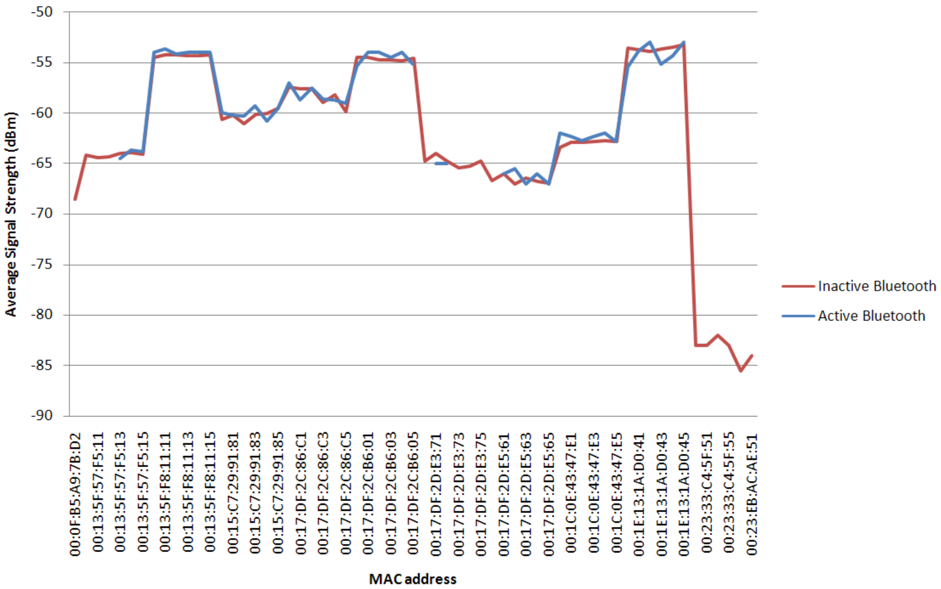


Figure 4.45: Average signal strength with Asus EeePC

Different access point MAC addresses are grouped on the X axis. The Y axis represents signal strength values of a particular access point averaged over 35 scans. Blue line represents average accuracy while having Bluetooth enabled.

The effect negative effect of scanning with both Bluetooth and Wi-Fi interfaces is best visible in the leftmost, middle and rightmost parts of the figure. We can see that if the wireless signal strength is in the range of -70 and -90 dBm – it will probably be blocked while performing Bluetooth and Wi-Fi scans at the same time. However the overall signal strength indication is not disturbed – signal strengths from the same access points tend to be very similar in both cases. The only negative effect that Bluetooth brings in this case is that it prevents some signals to be scanned but there is no major distortion in the measurement of signal strength.

Figure4.46 another Bluetooth and Wi-Fi interference measure. In this test we have recorded how often Bluetooth blocks Wi-Fi signal reception.

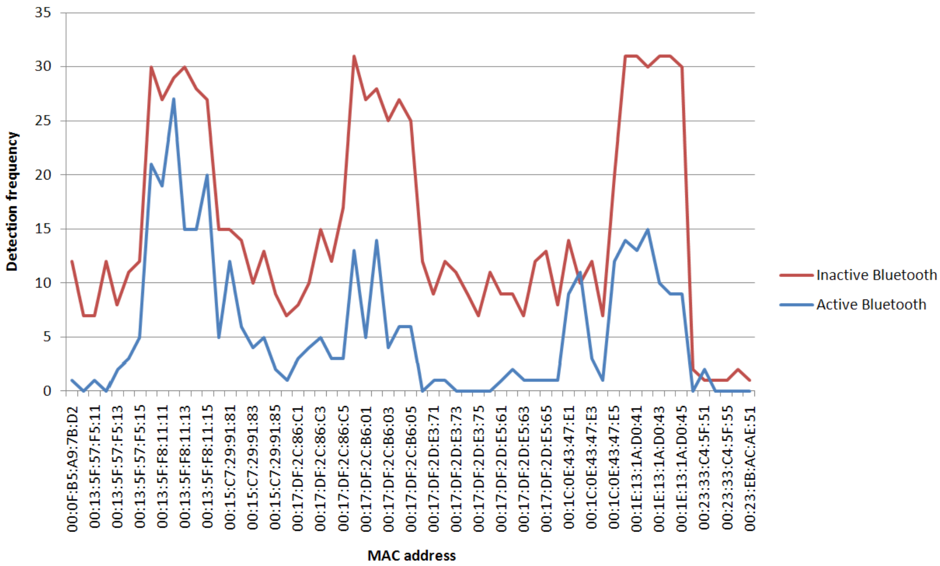


Figure 4.46: Detection rate Asus EeePC

As in the last test – we have made another 35 scans and measured how many times, out of 35, we could detect a particular access point.

Here - the interference problem is more obvious. We can clearly see that in almost every case scanning with having Bluetooth disabled resulted in a much better detection frequency.

In order to figure out how a different device would perform in the same situation we have conducted the same tests with Compaq nw8440.

Figure4.47 sums up the average signal strength indication.

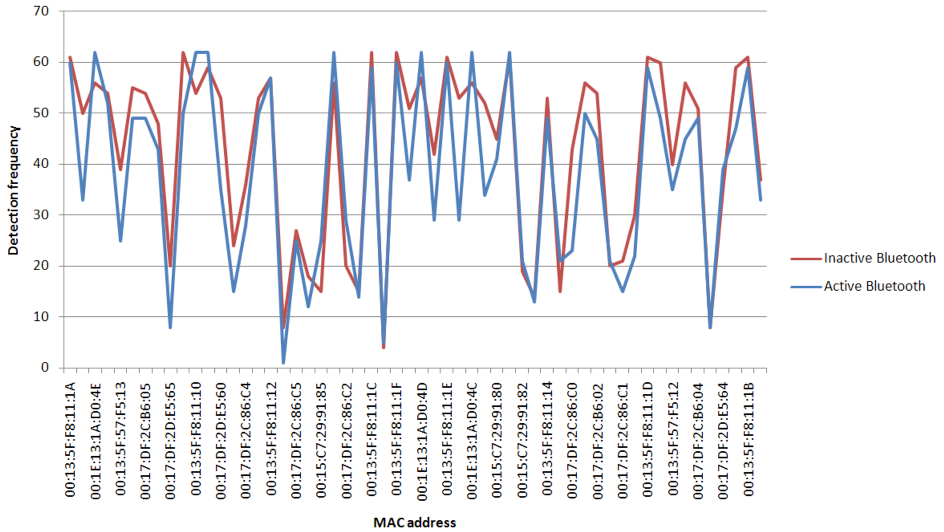


Figure 4.47: Average signal strength with Compaq nw8440

Clearly – this time the situation is different from previous one. We can see that in both scanning runs the results are more or less on the same level. The Bluetooth interference is much less noticeable using this machine.

Figure4.48 below just confirms that. The detection counter is again very similar. The difference in barely noticeable.

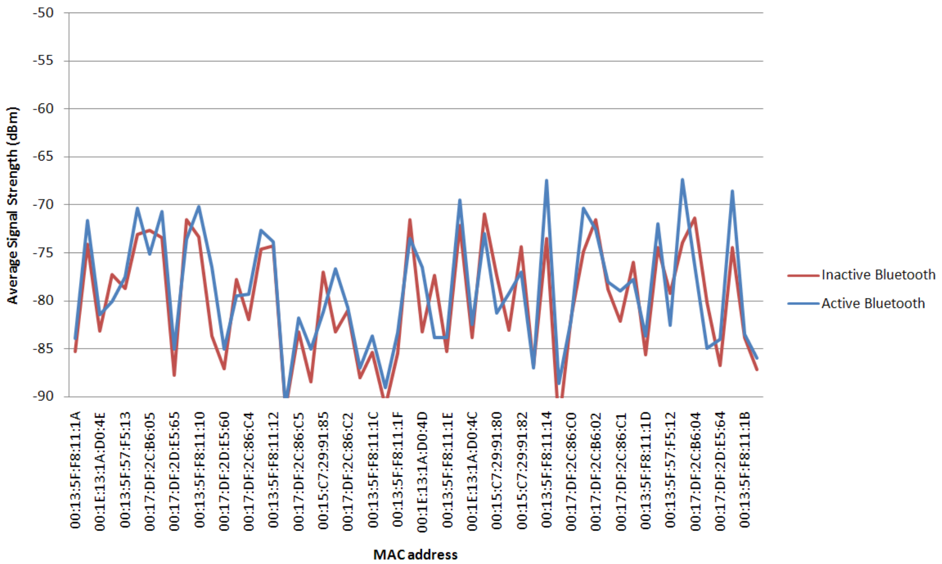


Figure 4.48: Detection rate with Compaq nw8440

It is worth mentioning that this device had both Wi-Fi and Bluetooth interfaces built in. Despite of this fact the interference between the two was not noticeable.

The only conclusion that we can make from these tests is that the results are highly dependable on the configuration of the machine. In some cases - performing simultaneous scans with both Wi-Fi and Bluetooth interfaces might cause an obvious interference. While another configuration might be completely immune to that kind of problems.

In general – it would be a better idea to use Bluetooth stations and make them scan for the clients that come close. That way clients would need to use only wireless network interface on their device. However as mentioned before we were not able to get the required equipment so we were used to use the alternative approach.

4.7.7.2 Using Bluetooth

In this section we present the results of our first real life test. In this first experiment we were using Bluetooth devices positioned in the same places as shown in the previous simulation experiment in section4.7.6 figure4.38. As before - the whole experiment area was divided into 4 regions.

Figure4.49 presents the results of this case.

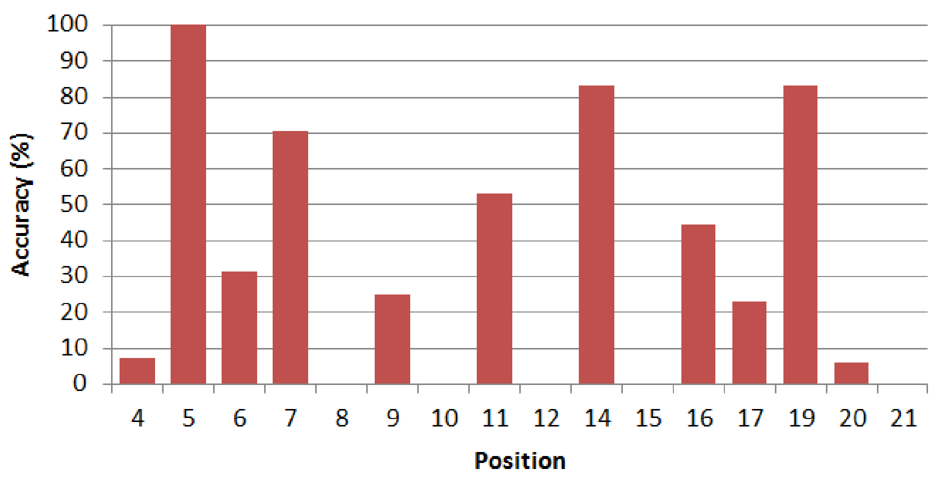


Figure 4.49: Accuracy while using Bluetooth

As expected – nodes 5, 7 and 19 - where our Bluetooth devices were positioned showed quite high accuracy measures compared with the overall average. The accuracy in all the other ones was clearly lower.

Average accuracy of this case is recorded at 33% level.

Because of the way we evaluate this test – average error distance is a much more interesting metric for us. Even if we get 0% accuracy we can still have average error distance of, say 5 meters, and that would indicate that the system is actually not that bad.

Figure4.50 presents the average error distance.

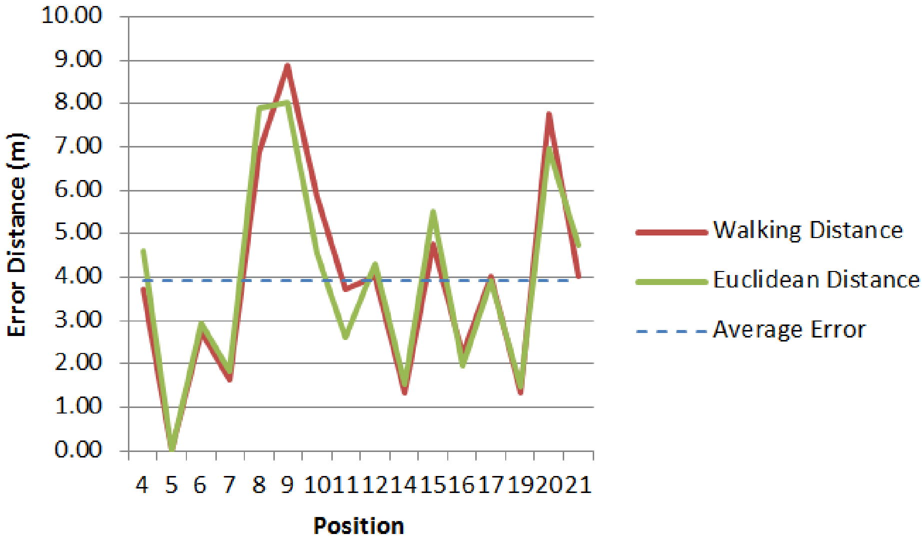


Figure 4.50: Error distance while using Bluetooth

As we can see the results here are very unstable and change from position to position. In some positions the distance is very minimal while in others it suddenly jumps up. In position 9 it reaches up to 8 meters. This can be explained by looking at how regions are arranged. Position number 9 belongs to the one of the longest regions. So when user entered this are the system was more sensitive to errors because of the fact that region contained some reference points that were separated by more than 10 meters from each other.

Average Euclidean error distance was very similar to average walking distance in all the positions. Overall average error distance of all the positions of this case is 3.92 meters.

4.7.7.3 Using graph model

In this section we present results of a very similar test to the one described above. The only difference is that this time we were using a graph model as describe in section 4.7.1. Based on our previous experience we have chosen the reachability parameter to be 3 as it yielded the best results in the simulation phase.

Figure4.51 shows the accuracy results of this case.

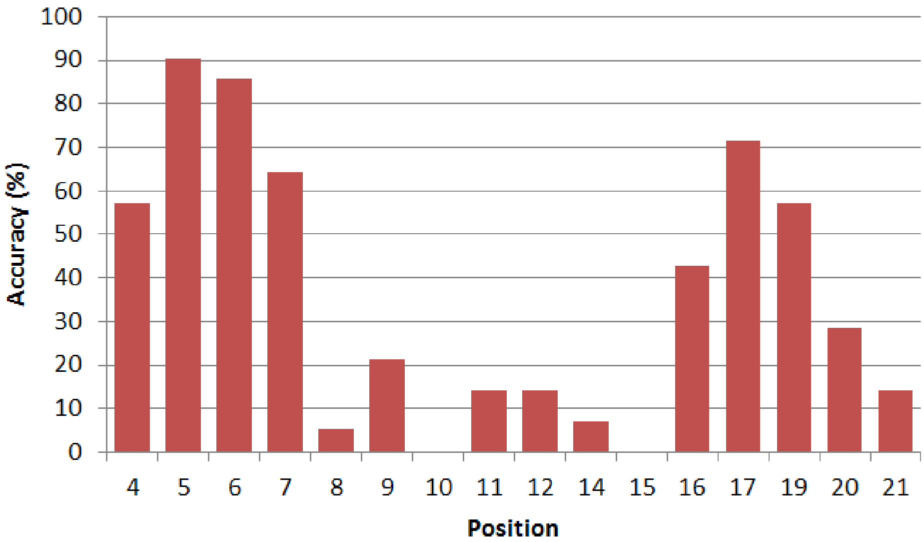


Figure 4.51: Accuracy while using graph model

As we can see from the figure above the accuracy results of this case tends to follow some trends. Accuracy reaches the maximal value of 90% in position 5. It then drops and later again increased around the area near position 17. The overall average accuracy in this test is 36%. It is almost identical to the previous case.

Figure4.52 shows the average error distance measurements.

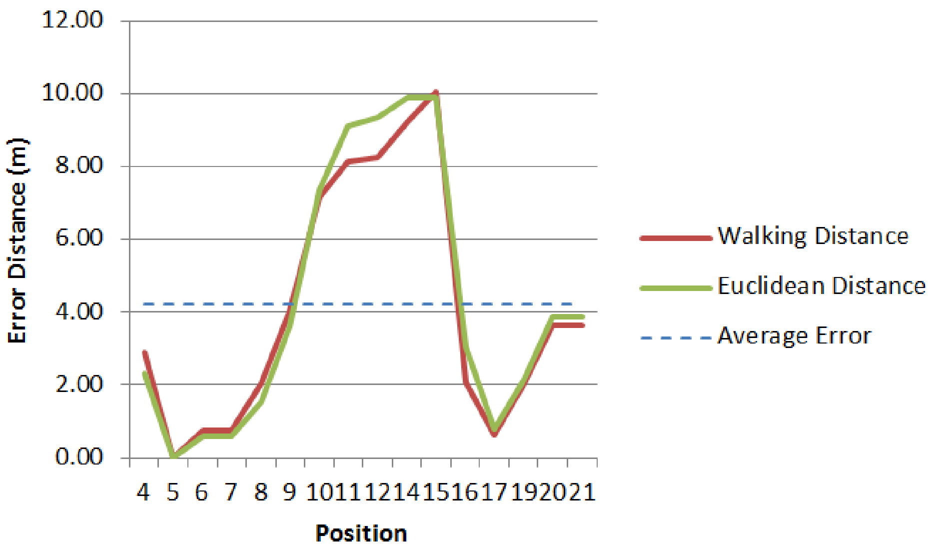


Figure 4.52: Error distance while using graph model

There is a clear indication that the area covering reference points from 10 up to 15 is the most difficult for the system to cope with. This can be related to the fact that this area has a fairly low Wi-Fi signal coverage compared with the rest of the building. Using the previous method the same area was the most problematic as well. The maximum average Euclidean error distance in this case was observed at positions 14 and 15 with the value of 9.91 meters. Average error distance stays at the 4.24 meter mark. This is a little worse than in the previous case.

4.7.7.4 Conclusions

The system's accuracy using the graph model is less "jumpy" compared with the Bluetooth approach. This behavior might be related to the fact that using Bluetooth we strictly define the available set of user's current positions according to what region he or she currently is. As we could see in some regions accuracy tends to be somewhat higher than in others. That explains why accuracy of this approach is so dissimilar in different positions. The results of using the graph model were more uniform – we could identify some trends and detect what areas caused the most accuracy related problems.

While using Bluetooth it would be a good idea to separate the region with the lowest accuracy into a few smaller ones. That would significantly lower the average error distance.

In general – we cannot state that one approach is better that the other. Decision to use one approach or the other depends on situation and current environment. It is also important to mention that Wi-Fi and Bluetooth signal interference issue may have had a negative effect on the results of experiments where Bluetooth devices were used.

Approach	Advantages	Disadvantages
Bluetooth	▶ Improved precision ▶ Maximum error distance can be tweaked ▶ Provides a framework for additional services and techniques ▶ Accuracy is less dependent on Wi-Fi ▶ Improved performance of the system	▶ Requires additional equipment and setup ▶ Very sensitive to possible Bluetooth device faults
Graph	▶ Improved precision ▶ Requires no additional equipment ▶ Improved performance of the system	▶ Results are solely dependent on Wi-Fi

Table 4.7: Summary of approaches

As shown in table 4.7 above both approaches improve the overall precision of the system. Using Bluetooth has more advantages but at the same time requires additional equipment. Furthermore – if one of the Bluetooth devices fails – it may significantly affect the overall results of the system. One of most important advantages of using Bluetooth over graph is that it enables us to pick the size of each region which in turn defines what the maximum possible error distance is. Another big advantage is that it enables a framework to develop some additional techniques such as those described in section 4.7.3. This approach is also more appropriate in situations where there are many clients with different device configurations as it is not solely dependent on one Wi-Fi infrastructure. Users with different NICs should be able to achieve better results by using this approach.

By using both approaches we were able to achieve the average error distance of approximately 4 meters. Knowing the typical precision of indoor positioning systems where error distance fluctuates between 1-6 meters based on [18], [20] and keeping in mind that our tests were carried out in a very dynamic environment this is a decent achievement.

Chapter 5

Conclusions

Having analyzed the results of experiments that were carried out throughout the project we can conclude that techniques described in this report help in making the indoor positioning system more robust even in environments where no consistent Wi-Fi signal strength can be expected throughout the longer periods of time.

Graph model

Incorporating the undirected graph structure into indoor positioning system resulted in slightly increased (5%) accuracy. However, a more noticeable improvement (17%) was observed in the average error distance. Even when the system still cannot estimate the correct position of the user – using this approach the estimated position is usually closer to the correct one. Another advantage of this approach is that by changing the reachability parameter we can limit the set of user's possible positions. This way it is possible to significantly increase performance of the system. Our location estimation procedure became more than twice as fast using this approach.

In general – we were not able to identify any of disadvantages of this approach. We expect that this approach would show even more improvement in real life cases where users tend to move a lot. In such cases this method should reduce frequent position estimation errors more noticeably. The fact that it improves both accuracy and performance of the system makes it applicable in most indoor positioning scenarios.

Combining different infrastructures

The idea of combining high range wireless infrastructure (Wi-Fi) together with low range (Bluetooth in our case) into a one integrated system showed a more significant improvement. We observed a 35% increase in accuracy and average error distance became approximately 5 times lesser.

While employing this approach in a real world situation we got very similar results compared to those that were achieved while using the graph model. This might be because of the fact that we were forced to run two simultaneous (Wi-Fi and Bluetooth) scans from the same client device which as documented in section 4.7.7.1 brought some interference issues. To avoid this issue we would suggest deploying Bluetooth stations and that way making them scan for the clients that come close. That would eliminate the requirement of forcing each client to do two simultaneous scans at the same time and also reduce energy consumption.

Another advantage of this approach is that it serves as a framework for additional techniques such as those explained in section 4.7.6.2 which helps to make indoor positioning systems more adaptable to different client device configurations and current environment.

Splitting fingerprints by time

Splitting current fingerprints into different sets according to time when fingerprint was collected is a helpful technique in cases where access point signal strengths vary during the course of the day. By using this technique we were able to reach a rather high accuracy level (96%). The average error distance was reduced to become more than twice as small. However this technique is intended to improve accuracy in inconsistent environments and may show different results in environments where signal strength does not change during longer periods of time.

History based approaches

According to the experiment results, methods that query history of a specific client while estimating its position should only be used in certain situations. It is because of the fact that the system cannot guarantee that those previously recorded positions are estimated correctly. That is the reason why we apply this approach only in those situations where indoor positioning system cannot determine a clear single candidate while choosing the most probable position.

As shown by tests in section 4.7.4.4, this approach may be helpful in cases where history of previous positions of the client is more or less accurate. However in cases where collected history is erroneous this approach is not applicable as it may even worsen the results.

Adapting to different network interfaces

Our effort to make indoor positioning system more adaptable to different client device configurations demonstrated improvement in cases where position fingerprints were collected with different devices. We were able to increase accuracy of the system by 28% and reduce average error distance to less than a half.

Future work

To further improve the ability to adapt to different network interfaces it would be interesting to experiment with Hyperbolic Location Fingerprinting [14] - storing Wi-Fi signal scan results as signal strength ratio of available access points. That might further improve indoor positioning system's performance when dealing with different network interface interfaces.

Using Bluetooth stations instead of Bluetooth-enabled devices is another possibility to reach further improvements. As mentioned in section 4.1.2 – this approach should help with minimizing Wi-Fi and Bluetooth signal interference problem. Otherwise – techniques explained in [6] can be employed trying to minimize the consequences of this interference issue.

Another interesting research area is to experiment with combining Wi-Fi with other low range wireless technology such as RFID or IrDA. That might as well help in solving wireless signal interference problem.

Appendix A

Figures

A.1 Access points in Cassiopeia

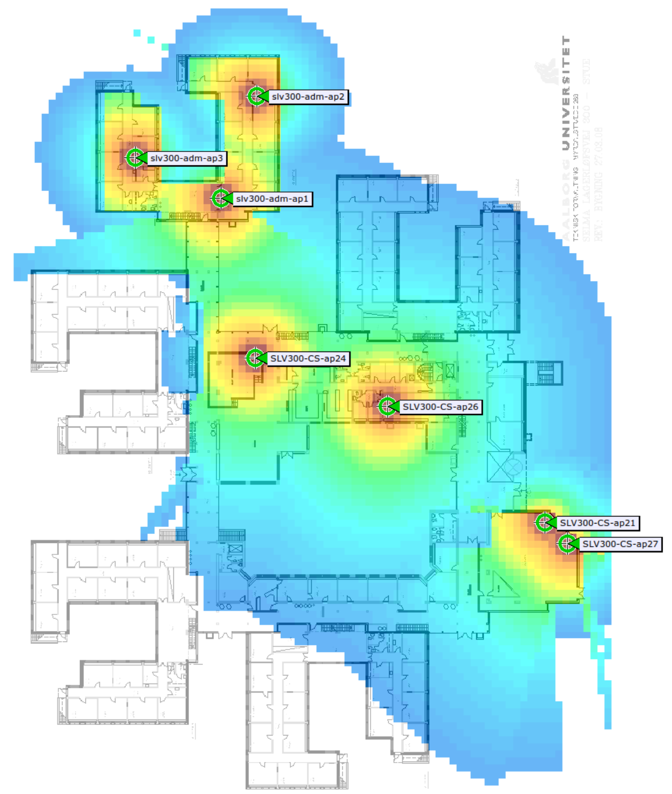


Figure A.1: Ground floor - where experiments took place

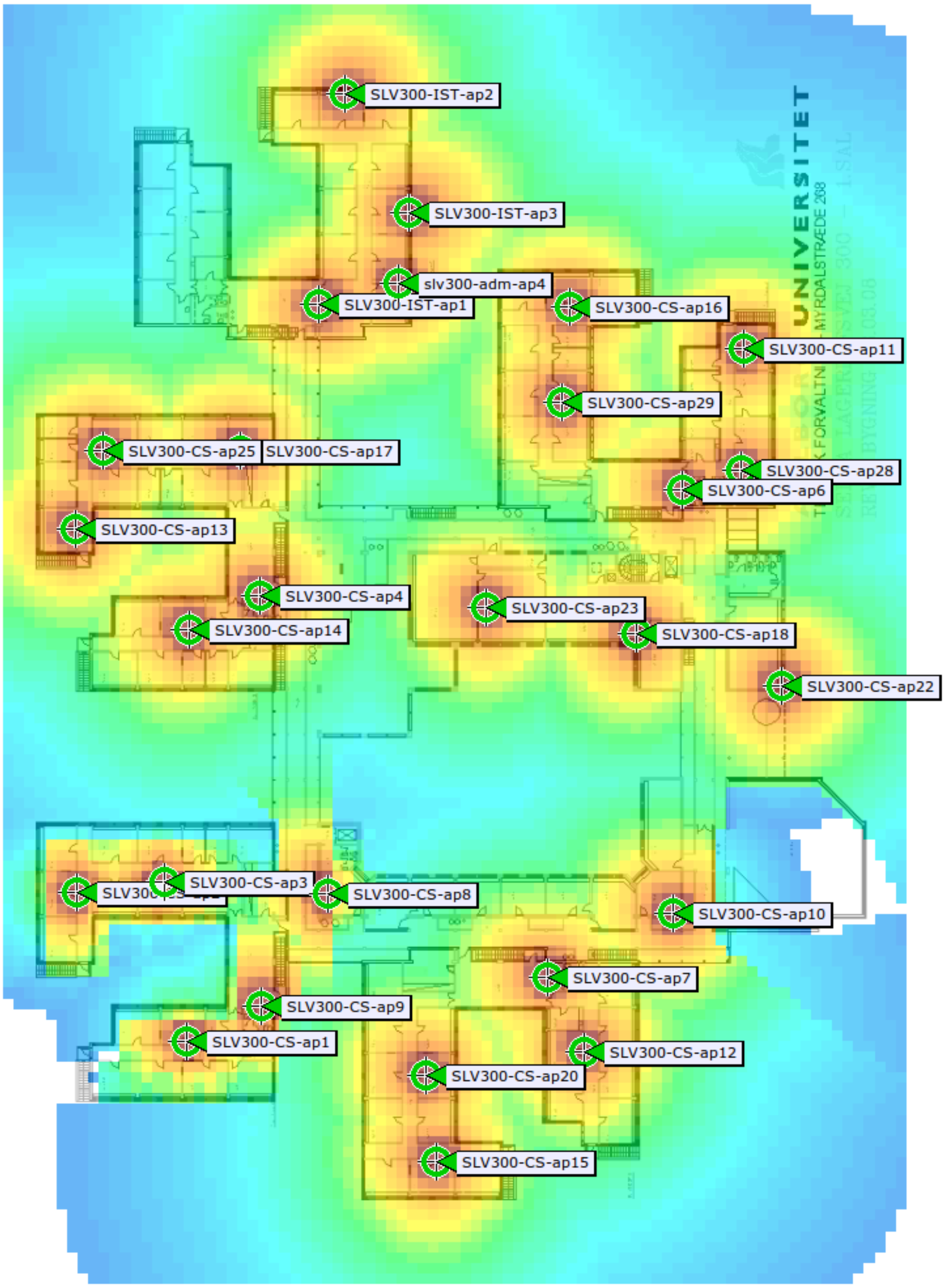


Figure A.2: First floor

A.2 Multithreading in our application

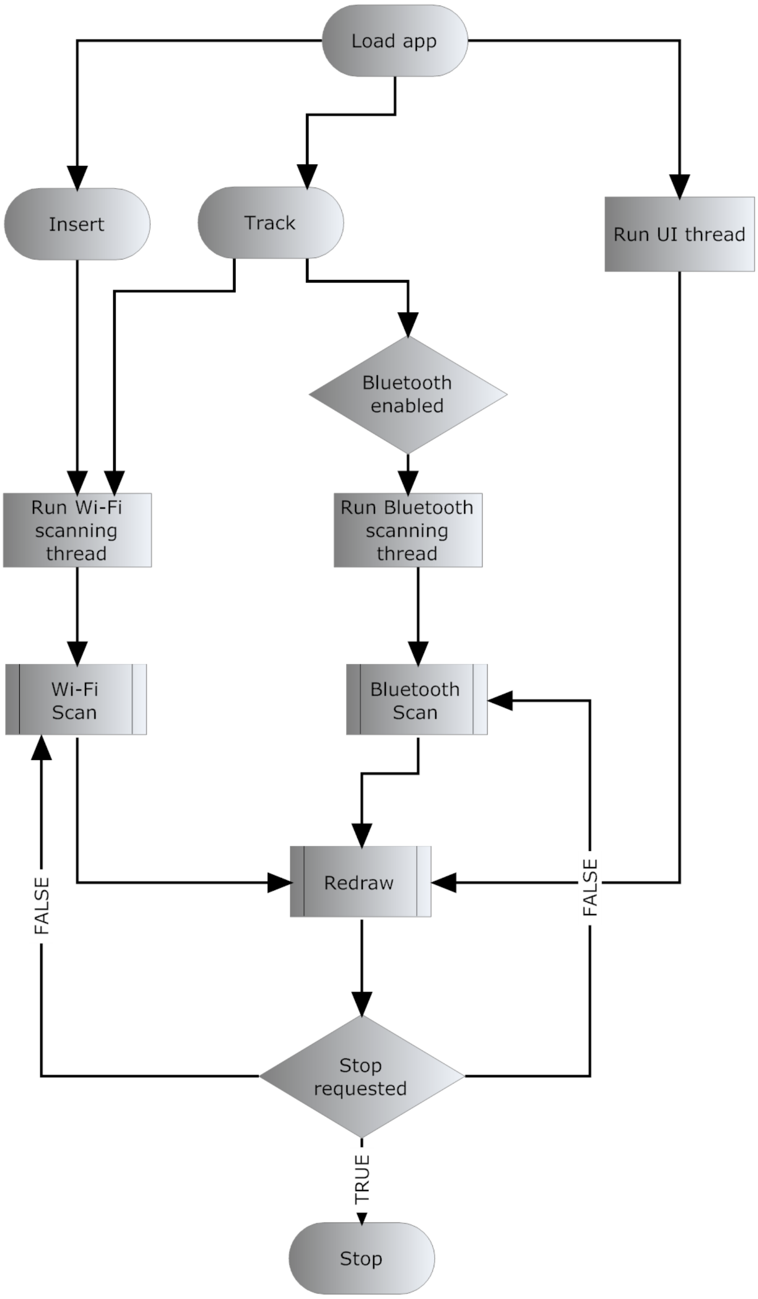


Figure A.3: Multithreading in our application

A.3 Fingerprint collecting phase

Figure A.4: Convenient way of collecting fingerprints

Appendix B

Terms and abbreviations

AP - Access Point

API - Application Programming Interface

BFS - Breadth-first Search Algorithm used in graph theory

BSSID - Basic Service Set Identifier, in an infrastructure BSS, the BSSID is the MAC address of the wireless access point [28]

dBm - power ratio in decibels referenced to milliwatts [27]

DBMS - Database Management System

ER diagram or **ERD** - Entity Relationship Diagram

Fingerprint - Vector of received signal strengths at particular position in indoor space [12]

GUI - Graphical User Interface

GPS - Global Positioning System

ID - Identification

IIS - Internet Information Services

IR - Infrared

IrDA - Infrared Data Association

MAC - Media Access Control address

MU - Mobile or Moving User

N, S, E, W - Directional Coordinates North, South, East, West

NDIS - Network Driver Interface Specification

NIC - Network Interface Controller also known as Network Interface Card

NNSS - Nearest Neighbor in Signal Space

PDA - Personal Digital Assistant

Radio Map - Fingerprint database

RF - Radio Frequency

RFID - Radio-frequency Identification

RSS - Received Signal Strength

RSSI - Received Signal Strength Indication

SQL - Structured Query Language

SS - Signal Strength

SSID - Service Set Identifier, is a name that identifies a particular 802.11 wireless LAN [28]

Wi-Fi - Wireless Fidelity

WLAN - Wireless Local Area Network

References

[1] Paramvir Bahl, , Paramvir Bahl, and Venkata N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. pages 775–784, 2000.

[2] Bluetooth.com. Bluetooth basics. `http://www.bluetooth.com/English/Technology/Pages/Basics.aspx`.

[3] AT&T Laboratories Cambridge.

[4] Cisco. Cisco wireless control system. `http://www.cisco.com/en/US/products/ps6305/`.

[5] Cisco. Wireless lan controller (wlc) faq. `http://www.cisco.com/en/US/products/ps6366/products_qanda_item09186a008064a991.shtml`.

[6] Mobilian Corporation. Wi-fi (802.11b) and bluetooth: An examination of coexistence approaches, 2001.

[7] I. Cubic, D. Begusic, and T. Mandi. Client based wireless lan indoor positioning. In *Telecommunications, 2005. ConTEL 2005. Proceedings of the 8th International Conference*, 2005.

[8] Infrared data association. Irda. `http://www.irda.org`.

[9] René Hansen and Bent Thomsen. Efficient and accurate wlan positioning with weighted graphs. In *MOBILIGHT*, pages 372–386, 2009.

[10] Christian S. Jensen, Hua Lu, and Bin Yang. Graph model based indoor tracking. In *MDM '09: Proceedings of the 2009 Tenth International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 122–131, Washington, DC, USA, 2009. IEEE Computer Society.

[11] Guang-yao Jin, Xiao-yi Lu, and Myong-Soon Park. An indoor localization mechanism using active rfid tag. In *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing -Vol 1 (SUTC'06)*, June 2006.

[12] Kamol Kaemarungsi and Prashant Krishnamurthy. Modeling of indoor positioning systems based on location fingerprinting, 2004.

[13] Mikkel Baun Kjaergaard. A taxonomy for radio location fingerprinting. In *LoCA'07: Proceedings of the 3rd international conference on Location-and context-awareness*, pages 139–156, Berlin, Heidelberg, 2007. Springer-Verlag.

[14] Mikkel Baun Kjaergaard and Carsten Valdemar Munk. Hyperbolic location fingerprinting: A calibration-free solution for handling differences in signal strength (concise contribution). Washington, DC, USA, 2008.

[15] Binghao Li, Jeffrey Kam, Jonathan Lui, and Andrew G. Dempster. Use of directional information in wireless lan based indoor positioning. In *in Symp. on GPS/GNSS (IGNSS2007)*, 2007.

[16] Binghao Li, James Salter, Andrew G. Dempster, and Chris Rizos. Indoor positioning techniques based on wireless lan. In *First IEEE International Conference on Wireless Broadband and Ultra Wideband Communications, Sydney, Australia, 13-16 March, paper 113*, 2006.

[17] Tsung-Nan Lin and Po-Chiang Lin. Performance comparison of indoor positioning techniques based on location fingerprinting in wireless networks. In *2005 International Conference on Wireless Networks, Communications and Mobile Computing*, 2005.

[18] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. 2007.

[19] In The Hand Ltd. 32feet.net - personal area networking for .net. `http://32feet.codeplex.com/`.

[20] R. Mautz. Overview of current indoor positioning systems.

[21] metageek. inssider. `http://www.metageek.net/products/inssider`.

[22] Microsoft. Native wifi. `http://msdn.microsoft.com/en-us/library/ms706556(VS.85).aspx`.

[23] Microsoft. Wlanscan function. `http://msdn.microsoft.com/en-us/library/ms706783(v=VS.85).aspx`.

[24] Kavitha Muthukrishnan, Maria Lijding, and Paul Havinga. P.: Towards smart surroundings: Enabling techniques and technologies for localization. In *In: Proceedings of the First International Workshop on Location and Context-Awareness (LoCA), SpringerVerlag*, 2005.

[25] NDIS.com. Ndis developer's reference. `http://www.ndis.com/`.

[26] A Zapater S.Feldmann, K Kyamakya and Z Lue.

[27] Wikipedia. dbm. `http://en.wikipedia.org/wiki/DBm`.

[28] Wikipedia. Service set (802.11 network). `http://en.wikipedia.org/wiki/Service_set_(802.11_network)`.

[29] Z. Xiang, S. Song, J. Chen, H. Wang, J. Huang, and X. Gao. A wireless lan-based indoor positioning technology. *IBM J. Res. Dev.*, 48(5/6):617–626, 2004.

[30] Da Zhang, Feng Xia, Zhuo Yang, Lin Yao, and Wenhong Zhao. Localization technologies for indoor human tracking. 2010.