

Bluetooth Indoor Positioning

Master of Computer Science

Submitted by
Anja Bekkelien

Supervised by
Dr. Michel Deriaz
Dr. Stéphane Marchand-Maillet
University of Geneva

March 2012

Acknowledgements

First of all, I would like to thank Dr. Michel Deriaz for his support, guidance and advice throughout this project, and Dr. Stéphane Marchand-Maillet for helping me with corrections of this report. I also thank Julien Pession for his valuable input and help on the implementation side, as well as all those who have supported me during the course of this master degree.

Abstract

There is an increasing demand for indoor positioning systems as location based services are becoming more and more widespread. However, this field of research has not yet achieved the same success as that of outdoor positioning. The objective of this project was to implement an indoor positioning system relying on Bluetooth technology, and to integrate it into the Global Positioning Module (GPM) developed at the Institute of Services Science at the University of Geneva. The system compares the signal strengths of surrounding Bluetooth devices to a database of measurements taken across the indoor area, in order to estimate the user's position. Through an evaluation of the system, an accuracy of approximately 1.5 meters has been obtained.

Contents

1	Introduction	1
1.1	Indoor positioning	1
1.2	Global Positioning Module	2
1.3	Scope	3
1.4	Thesis outline	3
2	Literature Review	4
2.1	System Architecture	4
2.2	Bluetooth device discovery	5
2.3	Positioning techniques	6
2.3.1	Trilateration and RSSI	6
2.3.1.1	Dead reconing	7
2.3.1.2	Neural Networks	7
2.3.1.3	Fingerprinting	8
2.3.2	Cell based method	8
2.4	Commercial Systems	9
2.4.1	ZONITH Indoor Positioning Module	9
2.4.2	Waspmote	9
2.5	Hardware	9
3	Modeling of the positioning system	11
3.1	Bluetooth technology	11
3.1.1	Received Signal Strength Indicator (RSSI)	12
3.2	Location Fingerprinting	12
3.2.1	The Offline Phase	13
3.2.2	The Online Phase	14
3.3	Positioning Algorithms	14
3.3.1	k-Nearest Neighbor	15

3.3.2	Naïve Bayes' Classifier	16
4	Architecture and Implementation	17
4.1	Architecture	17
4.1.1	GPM	17
4.1.2	GMA	17
4.1.3	LocMap	18
4.1.4	The Bluetooth Provider	18
4.2	Implementation	19
4.2.1	User Interface	19
4.2.1.1	CheckGPM	20
4.2.1.2	Interface for Radio Map Creation	20
4.2.2	Database	22
4.2.3	Improvements	22
4.2.3.1	Motion detection	24
4.2.3.2	Shortening the Bluetooth Inquiry Time	24
5	Evaluation	26
5.1	Scenarios	26
5.1.1	Hospital	26
5.1.2	Museum	27
5.1.3	Airport	28
5.2	Preliminary Test	30
5.3	Environment	32
5.4	Experiments	32
5.4.1	Precision and Accuracy of Positioning Algorithms	34
5.4.1.1	Results	36
5.4.2	Room-level precision	38
5.4.2.1	Results	38
5.4.3	Performance When Walking	39
5.4.3.1	Results	39
5.5	Comparison with other indoor positioning systems	41

6	Conclusions and Future Work	43
6.1	Conclusions	43
6.2	Future work	44
6.2.1	Evaluate beacon placement and fingerprint locations	45
6.2.2	Scalability	45
6.2.3	Improve accuracy	45
	References	47

List of Figures

4.1	GPM architecture.	18
4.2	Bluetooth provider architecture.	19
4.3	Sequence diagram for position estimation using the Bluetooth provider.	20
4.4	The user interface of the Bluetooth provider in CheckGPM.	21
4.5	User interface for radio map creation and positioning.	23
4.6	Entity relationship diagram of the GPM database.	24
5.1	The results of the preliminary test to evaluate signal variation in rooms of varying sizes.	31
5.2	Nokia BH-105 Bluetooth Headset.	32
5.3	The first floor of University of Geneva’s Computer Center.	32
5.4	One of the offices (left) and the corridor (right) used in the experiments.	33
5.5	The locations where measurements were taken during the creation of the radio map (screenshot from LocMap).	35
5.6	The accuracy (a) and precision (b) of k-NN and k-NN regression for different values of k	36
5.7	The accuracy (a) and precision (b) of k-NN (for $k = 9$), k-NN regression (for $k = 7$) and Naïve Bayes’ Classifier.	37
5.8	Experimental set-up for evaluating the room-level accuracy.	38
5.9	LocMap screenshot: fingerprint locations used for evaluating system performance when walking.	40

Chapter 1

Introduction

Bluetooth has emerged as a viable choice of technology in indoor positioning systems due to the increase in the number of Bluetooth-enabled devices over the last decade. Bluetooth was initially developed as a cable replacement technology but properties such as low cost and high availability has led to the development of several indoor positioning systems relying on Bluetooth signals.

1.1 Indoor positioning

The GPS technology has made outdoor positioning highly successful and it is now applied in a wide variety of products. Due to the signal attenuation caused by the construction materials of buildings, indoor positioning systems cannot rely on this technology and as a result, has not been equally successful. Several indoor positioning systems have been developed over the last decade, relying on a wide variety of technologies, including WLAN, infrared and ultra-sound among others, but there are still few commercial solutions available, and the ones that do exist are often costly and complex to install [13].

Similarly, a wide range of positioning techniques have been proposed, the most commonly used techniques being trilateration, triangulation and fingerprinting, see [12]. Still, the field of indoor positioning is not yet mature enough to propose a universal solution that works in any setting. Currently, the choice of technology and positioning technique depends on the requirements of the system. For instance, radio frequency based technologies such as IEEE 802.11 are inexpensive to deploy but has lower precision than many other kinds of technologies, whereas a system based on ultrasound, such as the Active Bat system¹, has a very high precision but is relatively expensive.

¹See <http://www.cl.cam.ac.uk/research/dtg/attarchive/bat/> for more details.

1.2 Global Positioning Module

In this project we seek to develop an indoor positioning system based on Bluetooth technology. This system will be integrated into the Global Positioning Module (GPM), a software developed internally by the Institute of Service Science at the University of Geneva, and which provides the user with a geographical position by transparently selecting the technology, or *position provider*, that is the most appropriate in any context. The software did not have a Bluetooth provider, which is the reason why this project focuses on the Bluetooth technology. GPM is developed for the Android platform, which places certain restrictions on the system architecture since the positioning must take place on the smartphone itself. In other words, it is the smartphone that is responsible for detecting Bluetooth signals and estimating a position.

GPM is used in several research projects that are focused on the quality of life of elderly people, more specifically, to improve their mobility by offering geographical positioning. This means that the module could potentially be used in many different contexts, such as a person's own home, airports, hospitals and so on. We therefore seek to develop a system that is as versatile as possible, so that it is able to cope with different types of environment.

To summarize, the project has the following requirements:

- Must be based on Bluetooth technology.
- Must work in all indoor environments in which GPM is used.
- Must use a centralized architecture where the smartphones pull information from surrounding beacons, as opposed to a server-client architecture where the beacons pull information from the nearby smartphones and communicate this information to a server.
- Must be compatible with the architecture of GPM, and be integrated into GPM.

In order to evaluate the system, we conducted several experiments where we investigated the accuracy and precision of the system under various conditions, and compared the results to those of other radio signal based systems. We were mainly interested in the following:

- Accuracy and precision when the user of the system is not moving

- The room-level accuracy
- How the system is affected when the user is walking

1.3 Scope

An important aspect of this work was to evaluate the existing techniques used in indoor positioning and to select the one that best fits our needs. This thesis is therefore not concerned with the development of new position techniques but rather the evaluation and deployment of existing ones. In particular, we evaluate several different machine learning algorithms for fingerprinting, as well as evaluate the system performance for different scenarios so that afterwards, one can easily judge whether or not the system can be used in a given setting.

1.4 Thesis outline

This report consists of six chapters which are organized as follows:

Chapter 1 introduces indoor positioning, the Global Positioning Module and the requirements and evaluation criteria of the implemented system.

Chapter 2 presents an overview of the existing indoor Bluetooth positioning systems and the various positioning techniques they propose.

Chapter 3 describes the model of the implemented system, in particular the chosen Bluetooth signal parameter and the positioning technique, as well as the machine learning algorithms that are applied.

Chapter 4 presents the architecture and implementation of the system.

Chapter 5 presents and discusses the results of the evaluation of the system along with a comparison to other indoor positioning systems.

Chapter 6 concludes this report and presents future work.

Chapter 2

Literature Review

A review of the current literature shows that indoor Bluetooth positioning is generally used for two purposes, namely for tracking of people or objects and for providing location dependent information. An example of the latter is described in [19], where the authors present a system for providing information and advertising to mobile phones with an accuracy of about 10 meters. Possible scenarios in which their system could be used are supermarket adverts and guided tours. A mobile guide system for positioning on the University of Mauritius campus is presented in [3]. Still, the majority of the research done so far has been focused on tracking.

2.1 System Architecture

Here we will describe the system architectures that are commonly used in these kind of systems. In order to track a mobile device, a fixed architecture of Bluetooth beacons is necessary. This architecture often consists of inexpensive commodity devices and is generally easy to deploy.

Several authors have accomplished positioning by making the mobile devices responsible for localization by detecting beacons. This has the advantage that the mobile devices do not need to be in discovery mode, hence maintaining privacy. Also, since the computations are performed on the mobile device itself, no data is stored on external servers.

Feldmann et al. describes such an architecture in [10], where they implemented a prototype tracking system on a PDA of the type Compaq iPAQ Pocket PC H3970 using Microsoft eMbedded Visual C++. The system reads the Received Signal Strength Indicator (RSSI) from Bluetooth-enabled notebooks which are distributed inside a 46 m² laboratory.

Similarly, [5] implements a system where localization is done on handhelds such as mobile phones and PDAs. The author deployed beacons consisting of cheap USB Bluetooth adapters connected to a power source, worth about 5 USD each.

[11] describes a system where they reproduced a person's movement in a 3D environment. They used an HTC HD2 mobile phone to track the person by having it detect Bluetooth devices placed in the area. An initial calibration, where the devices to be used during tracking are registered along with their distances, is necessary, and this allows any kind of Bluetooth-enabled device to be used as beacons.

Another option is to have the beacons scan for mobile devices. This requires a distributed architecture where the beacons report information about the detected mobile devices to a server. The advantage is that there is no need for specialized software on the mobile devices and it does not require active participation from the users.

The Bluepass system [8] is an example of this. The goal of Bluepass is to allow users to locate and to be located by other users with the help of mobile devices. It is based on a server-client architecture where the detection of Bluetooth devices is done by a program running on PCs with USB Bluetooth adapters. The program is continuously scanning for mobile devices, and maintains a connection to the server, to which it sends the RSSI and address of the devices it has found. The server is responsible for calculating the position. Users are required to log in to the system via a Bluetooth-enabled device on which a mobile application is running and connected to the server. This gives them access to a map showing the position of all the users in the system.

BIPS [2] is another example. It is an indoor positioning service for tracking mobile users based on a central server and Bluetooth enabled workstations connected to the Ethernet LAN, which track the users who have logged in to the system. Similarly to Bluepass, the users can query the server for the position of other users.

The already mentioned mobile guide system for positioning on the University of Mauritius campus also uses this kind of architecture.

2.2 Bluetooth device discovery

A Bluetooth device detects other devices by executing the inquiry protocol. Devices within its range that are set to "discoverable" will respond by identifying themselves. This is a rather slow procedure that can take up to 10.25 seconds to complete and

prematurely aborting the procedure entails the risk of not detecting all devices. Interestingly, the authors in [15] conducted an experiment in which they investigated the effect of varying the inquiry time period, and obtained good results with values as low as 3.84 seconds. Most systems rely on the inquiry protocol for positioning, such as [4] which uses the inquiry response rate to localize devices.

Hay and Harle proposes an alternative in [14], where they explore *connection-based* tracking. This method is based on having a host connect directly to a mobile device instead of using the slow inquiry procedure. The advantage is that one can quickly connect to any given device. They make use of the low-level Asynchronous Connectionless Link (ACL) connection, which is sufficient for tracking purposes and does not require authorization. In order to locate a device the hosts would simultaneously try to connect to it. As the authors point out, this does not scale and they suggest to monitor the movements of the user and use this information to limit the search to only those hosts which are within the reach of the user. While this method has the advantage of low connection times (up to 1.28 seconds) the drawback is that the tracked devices must be known in advance. The authors conclude that the method has potential even though it has disadvantages which does not make it the perfect solution for tracking.

2.3 Positioning techniques

This section presents the various indoor positioning techniques that are appropriate for Bluetooth devices. The majority of the available research is focused on trilateration using the RSSI signal for calculating distances although several fairly recent articles have explored a cell based approach. Research is also being done on fingerprinting techniques.

2.3.1 Trilateration and RSSI

Trilateration is a way of determining a location by measuring the distance from a mobile device to three other beacons. As there is no direct way of measuring distance using Bluetooth, these methods rely on the various signal parameters defined in the Bluetooth specification, such as the Received Signal Strength Indicator (RSSI) and Link Quality (LQ), in order to infer the distance between two devices. [15] provides a thorough evaluation of all the signal parameters and how suitable they are for localization purposes. They conclude that it is not possible to obtain accurate distances using signal parameters.

Yet, several authors have successfully implemented indoor localization systems based on signal parameters [14] [11] [18] [10], most of them relying on the RSSI. This is due to the fact that in most cases, only room-level localization is required, hence a highly accurate measurement of distances is not necessary. It seems to be the agreed consensus that a finer granularity than room-level is difficult or impossible to obtain using signal parameters.

In [9], D. C. Dimitrova et al. investigates the accuracy of the inquiry-related RSSI (available from Bluetooth 1.2 onwards) when used to determine in-room and out-of-room locations. The inquiry-related RSSI is considered more reliable than RSSI because it is not affected by Bluetooth power control. The authors installed several beacons, spaced two meters apart, inside and outside of a room and tracked the positions of four different types of smart phones when moved inside and outside. The beacons were OveroFire gumstix nodes and they used the following mobile phones: an HTC Desire, an HTC Wildfire, an iPhone and an LG e900. They found that room-level positioning was indeed feasible using this method, but a finer granularity (a few meters) was difficult to obtain as there is no deterministic relationship between the inquiry-related RSSI and the actual distance. There also seemed to be certain device specific differences between the different brands of mobile phones.

Effort is being made in order to improve the accuracy of signal parameters for triangulation, as is described in the following sections.

2.3.1.1 Dead reconing

The afore mentioned 3D environment tracking system in [11] also uses RSSI and triangulation for positioning. In order to deal with the slow update rate caused by the inquiry protocol delay and make the positioning more accurate, they make use of the devices' built-in accelerometer and digital compass in order to keep track of the user's position while waiting for the next update. Their proposed solution includes an initial calibration phase where the devices that are to be used must be registered, as well as the RSSI and distance between them. The resulting system was able to obtain fairly accurate positioning (< 1 meter when the accelerometer is correctly configured).

2.3.1.2 Neural Networks

There is ongoing research into applying neural networks in indoor Bluetooth positioning in order to improve the accuracy of signal parameters, as is shown in [1]. Here, they address the issue of changes in the RSSI values due to user orientation. Their

approach is based on fingerprinting and require two phases: an offline phase where multiple neural networks are trained using collected RSSI values, and an online phase where the system is actually being used. A neural network is selected based on the orientation of the mobile device, which is found using a built-in compass. With this method the authors were able to increase the accuracy to 0.5 meters. The drawback of neural networks is the computationally heavy training phase, in particular for large training sets, that must be done beforehand, and which might not be desirable for an application developed for smartphones.

2.3.1.3 Fingerprinting

Subhan et al explores the possibility of improving the inquiry-related RSSI measurements using fingerprinting in combination with lateration in [20]. By applying a gradient filter to the measured inquiry-related RSSI values they were able to reduce the average error by 45%, from 5.87 meters to 2.67 meters.

2.3.2 Cell based method

Cell based methods rely only on the visibility of Bluetooth beacons for determining the position of a device. These methods divide the indoor area into sections according to the range of a given beacon, meaning that sections can overlap each other. The position of a device is inferred from which beacons the device can see and which beacons it cannot see. Cell based localization has been used in [6] [5] [7].

In [6], the environment is modeled using a location graph where the vertices represent locations and the edges represents the connection between locations. Information about visibility and non-visibility of beacons is then used to find the position of a device in the location graph. The focus of their work is to find a work-around for the delay associated with the Bluetooth inquiry protocol. They chose to use the paging protocol. This way individual beacons can be probed directly, which gives faster localization provided that one has a good idea of which beacons to probe. The authors presents an algorithm that finds a minimum-cost beacon-probing plan which is used for selecting the beacons that are to be probed.

[7] presents an inexpensive cell based Bluetooth positioning system, in which they wrapped beacons made of off-the-shelf hardware using cardboard boxes and foil tape in order to obtain a range of 2-3 meters. Each beacon had a total cost of about 20 USD. A test application that continuously scans for known beacons was deployed on Audiovox SMT5600 mobile phones, and beacon visibility was used to determine their positions. The authors conclude that the 2-3 meter range gave satisfactory results.

2.4 Commercial Systems

This section provides a brief overview of the only two, to our knowledge, commercially available systems which uses Bluetooth. One can only speculate why there are so few, but Bluetooth properties such as the long discovery delay and the lack of proper beacon hardware might be part of the reason. Additionally, buildings generally already has a Wi-Fi infrastructure in place, which could render Wi-Fi based systems more commercially interesting since the client would not have to install new infrastructure.

2.4.1 ZONITH Indoor Positioning Module

The ZONITH Indoor Positioning Module [21] is offered by Teldio and is a Bluetooth based localization system for monitoring employees that are working in hazardous environments. The system consists of ZONITH Bluetooth Position Beacons that are distributed throughout the area and are capable of tracking any Bluetooth-enabled mobile device. The beacons are connected to the Ethernet and allow the user to adjust the signal power in order to adjust the granularity of the positioning system.

The positioning system is integrated with Teldio's ZONITH Alarm Control System which is responsible for alerts in case of emergencies, for instance if the systems's lone worker protection is enabled.

This system will automatically track any Bluetooth device that enters the area and no modification of the hand held devices are necessary.

2.4.2 Waspnote

Libelium produces the programmable Waspnote Bluetooth Radio¹ for device discovery which is a module to be mounted on their Waspnote sensor board. The Bluetooth Radio is designed specifically to detect a large number of devices (250) in one single inquiry, and has been used as part of their resent Traffic monitoring platform.

2.5 Hardware

As this chapter shows, there are very few commercial systems that offer Bluetooth indoor positioning. Even simple Bluetooth beacons for positioning purposes are not available on the market today. Other devices that are originally designed for other purposes than positioning could potentially be used as beacons (eg. headsets, GPS senders) but they all suffer from a relatively short battery lifetime which renders them

¹<http://www.libelium.com/products/waspnote>

impractical for positioning purposes. It is therefore necessary to build customized beacons.

There are generally two ways in which a beacon can be constructed. The most common is to equip a computer with a Bluetooth dongle. This has the advantage that it allows existing infrastructure to be reused, since many indoor areas already has a computer in most rooms, such as universities and office buildings. An inconvenience with this approach is that one has to make sure that the computers are not turned off and that the dongles are not removed.

The second approach is to use USB hub connected to a power outlet together with Bluetooth dongles. This has the advantage that it does not rely on any infrastructure and is easy to deploy. The disadvantage comes with the fact that dongles need to be initialized before they can be used. It is enough to connect the USB hub and dongle to a computer once when the system is started up, however, this procedure will have to be repeated in case of power failures. This could be inconvenient if the system consists of a great number of units.

For the evaluation of our implementation, we have chosen to use a collection of class 2 Bluetooth headsets. Although their battery lifetime is low, it is still sufficient for our tests, and we avoid having to rely on the availability of power outlets as would be the case with USB hubs.

Chapter 3

Modeling of the positioning system

This chapter describes the Bluetooth technology in more details and discusses how it can be used in an indoor positioning system. It then goes on to describe the fingerprinting technique and the positioning algorithms that has been implemented.

3.1 Bluetooth technology

Bluetooth is a technology that allows electronic devices to communicate without wires. It was designed for low power consumption and is based on low-cost transceiver microchips. Bluetooth communicates using radio waves with frequencies between 2.402 GHz and 2.480 GHz, which is within the 2.4 GHz ISM frequency band, a frequency band that has been set aside for industrial, scientific and medical devices by international agreement. The Bluetooth specification was conceived in 1994 and is now managed by the Bluetooth Special Interest Group (SIG).

Bluetooth is divided into three classes, each of which has a different range, as shown in table 3.1. This range can potentially be affected by the surrounding environment, as the signals are susceptible to propagation effects. This is especially true in an indoor environment. Although class 3 devices would be ideal for indoor

Class	Range
Class 1	100 m
Class 2	10 m
Class 3	5 m

Table 3.1: Bluetooth range.

positioning purposes because of the small range, such devices are very uncommon and the vast majority of available devices are of class 2.

A device may connect to up to seven other devices simultaneously by setting up an ad-hoc network in which the device that initiates the communication acts as a master. Such networks can be set up without the need of user interaction. In order to avoid interference between devices, Bluetooth uses a technique known as *spread-spectrum frequency hopping*, in which a device will transmit on one of 79 randomly selected frequencies, changing from one frequency to another 1600 times per second. This makes it unlikely that two devices will transmit on the same frequency and therefore minimizes the risk of interference.

Before setting up a network, a device must perform an inquiry in order to discover other devices that are within its range, a rather lengthy process that may last for up to 10.24 seconds. Any device that is in *discoverable* mode will then reply by sending back information about itself, such as its name and address. The two devices can subsequently pair with each other, a process in which they create a common *link key* that is stored on each device. This is required by many services in order to increase security.

3.1.1 Received Signal Strength Indicator (RSSI)

Generally, the RSSI is a measurement of the strength of an incoming radio signal. It is a relative indicator and its units are arbitrary, but the higher the value of the RSSI, the stronger is the signal. In Bluetooth, the RSSI is used to tell whether the received signal is within the Golden Receiver Power Range (GRPR), which is the name used to describe the ideal range of incoming signal strengths. The RSSI is measured in dB, and a signal strength within the GRPR results in an RSSI of zero dB. A positive or negative RSSI indicates that the signal strength is above or below the GRPR, respectively. The Bluetooth specification does not specify the upper or lower limit of the RSSI but simply states that it must be possible to tell whether the incoming signal is within, above or below the GRPR, hence this value is device specific.

The RSSI of a Bluetooth device is obtained by starting the inquiry procedure from a second device. The RSSI will then be included in the first devices' response to the inquiry, meaning that it is not necessary for two devices to actually be connected or even be paired.

3.2 Location Fingerprinting

There are mainly two techniques that are used in Bluetooth indoor positioning today: triangulation and fingerprinting. The fingerprinting technique has been chosen for this

project, as the literature review shows that the Bluetooth signal parameters which are used in triangulation are not sufficiently accurate, see [15].

Location fingerprinting is a technique in which a location is identified by a record of radio signals. A major advantage of this technique is that if an appropriate radio infrastructure is already in place where the positioning system will be deployed, it can be reused, which avoids the cost of installing new infrastructure.

As stated in the literature review, techniques that derive a distance directly from the signal strength are susceptible errors due to the fact that there is no direct relationship between the signal strength and the distance. Such methods include triangulation and lateration, among others. The fingerprinting technique completely avoids this problem, as it is not at all concerned with the distance, but rather tries to obtain a unique combinations of RSSIs that distinguishes a location from all other locations. In principle, any Bluetooth-enabled device can be used as a beacon since differences in the emitted signals will have no bearing on the accuracy of the position estimation.

The fingerprinting technique is based on a radio map, which is a collection of *fingerprints*. A fingerprint is a set of radio signals measured at a particular location, in which each signal is associated with the device from which it was emitted. This technique comprises an offline phase, in which the radio map is created, and an online phase, where the actual position estimation takes place. The following sections provides a more detailed description of the two phases.

3.2.1 The Offline Phase

Consider a positioning system deployed on a single floor, consisting of N Bluetooth beacons that provide a complete signal coverage of the area. We define a grid over the area, which limits the possible positions to the intersection points of the grid. At each such point, we create m fingerprints that are stored in a database which is used as the radio map, and associate each of them with the point at which they were created. Each point is denoted by its (x, y) coordinate, and considering that there are k points on the x-axis and l points on the y-axis, this gives a total of $m \times k \times l$ fingerprints. The radio signals on the same locations are affected by the orientation of the human body [1], and it is therefore necessary to create fingerprints in several directions at the same point in order to account for this. Additionally, it is necessary to create enough fingerprints at a point to have a statistically valid sample of measurements.

In a real-life situation there might be points on which fingerprints are not created, either by choice or because it is not possible, for instance areas that are covered by furniture or other obstacles.

3.2.2 The Online Phase

The online phase is where the position estimation takes place. The fingerprints are represented as vectors, where each element i contains the RSSI received from beacon i . The elements corresponding to beacons that were not visible from the given location are assigned a value of zero. In order to estimate a position, the smartphone does an inquiry in order to obtain the RSSI of all visible beacons. This sample vector is then compared to the pre-recorded fingerprints in the database, and the position associated with the fingerprint that is the most similar to the sample vector is chosen as the estimated position.

3.3 Positioning Algorithms

Several algorithms have been proposed for predicting positions from a radio map. Kjærgaard [16] categorizes these algorithms into two groups, probabilistic and deterministic, and provides an overview of the algorithms that have been applied in this area. Common to all these algorithms is that they are based on a training set, or a collection of data samples, on which they have to be trained before they can provide an output. When used in the fingerprinting technique, this training set equates to the radio map.

The probabilistic algorithms considers the samples to be random values drawn from a distribution based on a known model, such as the Gaussian, and which is defined by a small number of parameters, such as the mean and variance. The training then consists of estimating these parameters from the sample set, after which the whole distribution is known. These parameter estimations are applied onto the model which gives us an estimated distribution which we can use to classify an input. The advantage of such methods is that they reduce the problem of estimating a probability distribution function down to simply estimating a few parameters. The disadvantage, on the other hand, is that it assumes that the model is valid for the entire sample space. If this is not actually the case, the output will have large errors.

Deterministic algorithms uses the sample set directly and does not make any assumptions about the underlying density of the data. The only assumption that is made is that similar inputs give similar outputs. The problem therefore consists of

finding the samples in the training set that are the most similar to the input using some kind of distance measure. A commonly used measure is the Euclidean distance. A consequence of this approach is that the complexity of the algorithm is directly related to the sample size, since the input sample has to be compared to each training sample.

A large portion of the research in indoor positioning using Bluetooth has applied a probabilistic algorithm, yet variations on k-Nearest Neighbor as well as the SVM algorithm are frequently seen. As nearly any machine learning algorithm can be applied for this purpose, it is impossible to evaluate all of them. We have instead focused on the ones that are most commonly used, all of which gives fairly good but also fairly similar results. The ones we have chosen are weighted k-Nearest Neighbor, regression based on k-Nearest Neighbor and the Naïve Bayes classifier.

3.3.1 k-Nearest Neighbor

k-NN is a deterministic classifier that compares its input against the entire training set at runtime, and can therefore be computationally intensive. It is, however, known to give good results and is simple to implement.

The output is based on a comparison between an input vector and the training set using a distance measure such as the Euclidean distance, in the case of continuous values. The k instances of the data set that are the most similar to the input (have the shortest distance to the input) are selected, and those instances vote on the class of the output. Each instance has one vote, and votes on its own class. It is therefore the class that has the majority among the k nearest neighbors of the input that is selected as the output. For this reason, k is often chosen odd to avoid draws between classes.

A drawback of this algorithm is the fact that all k neighbors have the same vote, independent of their distance to the input. The algorithm can be improved by weighting the neighbors by their distance to the input, and the inverse of the distance is often used. This means that the closer an instance is to the input, the more its vote influences the result.

k-NN can also be used for regression, by having it return the average value of the k neighbors.

3.3.2 Naïve Bayes' Classifier

The Nave Bayes classifier is based on Bayes theorem, which gives us the probability of C given x :

$$P(C|x) = \frac{P(C)P(x|C)}{P(x)}$$

where the different elements of the formula are denoted as follows:

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

In the case of the fingerprinting technique, this translates into the probability that fingerprint x belongs to class C , where the class is a position, described by its coordinates, and x is a vector where each element contains an RSSI reading.

The classifier is called naïve because it makes the assumption that the values in the input vector are all independent of each other. This makes it easy to calculate the conditional probability $P(x|C)$, which simply becomes the product of the probability of each element in x given class C :

$$\prod_i P(x_i|C)$$

Because the *evidence*, $P(x)$, only serves to normalize the result, it is usually ignored and the formula is reduced to

$$P(C|x) = P(C) \prod_i P(x_i|C)$$

In order to classify x we use the *maximum a posteriori* decision rule, which simply means that we compute the above formula for each class and select the class which has the highest probability as the output:

$$\text{choose } C_i \text{ if } P(C_i|x) = \max_k P(C_k|x)$$

This implies that the classifier will give the correct result as long as the correct class has the highest probability, even if the probabilities are not distributed correctly among the classes.

Chapter 4

Architecture and Implementation

Having already discussed the methods and algorithms on which the indoor positioning system is based, this chapter presents the architecture and implementation of the Bluetooth GPM provider as well as the projects in which it has been integrated.

4.1 Architecture

Since the provider was to be integrated into the already existing software developed at the Institute of Services Science, this section gives a brief overview of this software which consists of three separate projects: Global Positioning Module (GPM), Global Map API (GMA) and LocMap, all developed for the Android platform.

4.1.1 GPM

GPM is designed as an API that can easily be integrated into applications that need to provide position information. The API is based on a collection of position providers which each provides position information using different types of technology, such as WiFi, dead reckoning, GPS, barcodes etc. Figure 4.1, taken from the documentation of GPM v1.2, shows the general architecture of this project. The kernel provides the abstract class `Provider` to which all position providers must adhere. This class is then used by class `GPM`, which provides the methods available to the code that interacts with the API. `CheckGPM` is a small application designed for developers to test the respective position providers.

4.1.2 GMA

GMA does not deal with positions but rather provides transparent access to different types of maps, more specifically Google Maps and Open Street Map. It also provides

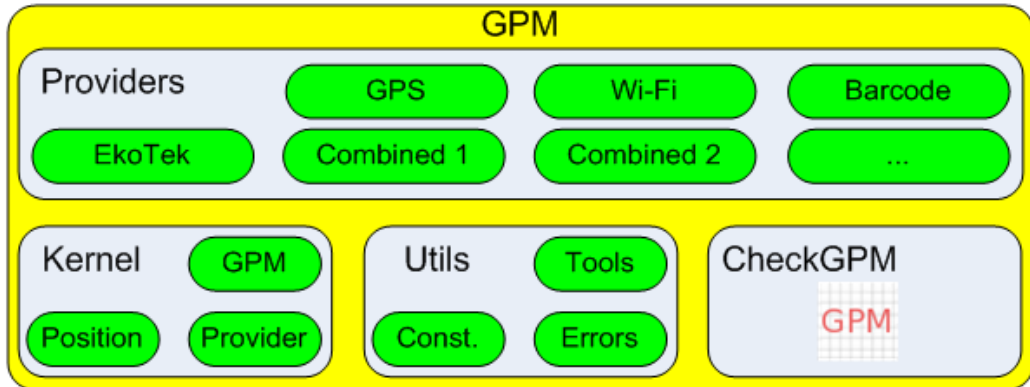


Figure 4.1: GPM architecture.

functionality for creating basic types of geographical features such as points, lines and polygons, independently of the the map type.

4.1.3 LocMap

The purpose of LocMap is to demonstrate the features of the two aforementioned projects. It is therefor a light application that simply interacts with the APIs provided by GPM and GMA in order to present their features in a concise way. Among other things, it allows to change between different position providers and to configure their respective settings.

4.1.4 The Bluetooth Provider

A new package, `bluetooth`, was added to GPM and it consists of four principal classes as well as two packages containing the implementations of the positioning algorithms and the database access layer. This architecture is shown in figure 4.2. The class `Bluetooth` has the main responsibility of starting the Bluetooth inquiry and estimating a position based on the outcome from the inquiry as demonstrated by the sequence diagram shown in figure 4.3, which outlines the actions taken during the online phase of the system.

Bluetooth inquiries are initiated when the mobile phone detects movement, which is managed by the `MotionDetector`, a feature implemented in order to save battery power. It listens to the build-in accelerometer, and inquiries are then launched continuously until the `MotionDetector` notices that the mobile phone is no longer in motion. While the mobile phone is moving, `Bluetooth` starts asynchronous inquiries by calling the `BluetoothScanner`, which provides it with information about all the

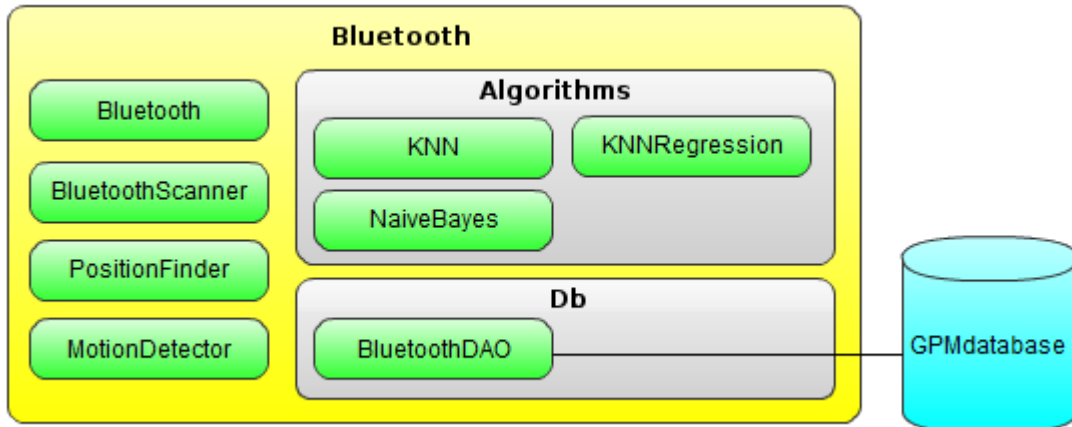


Figure 4.2: Bluetooth provider architecture.

devices that are within range and in discovery mode. Once the asynchronous call terminates, `Bluetooth` calls the `PositionFinder` in order to obtain a position estimate based on the returned devices. This class will use one of the positioning algorithms as well as the radio map stored in the database (not shown in the sequence diagram) in order to come up with an estimation, which is returned to `Bluetooth`.

The `Bluetooth` class has a similar role during the offline phase, however, each inquiry is explicitly initiated by the user and not by the detection of movement. When an inquiry has finished, instead of running one of the positioning algorithms, the resulting fingerprint is stored directly in the database.

Another important aspect of the provider was the selection of beacons. It is necessary to specify which beacons that are to be taken into account by the system in order to avoid that arbitrary devices, such as peoples mobile phones, headsets, laptops etc. are included. Including such devices would break the radio map once the device is moved to a different location. The provider therefore takes this list into account during the position estimation and excludes any device which is not on the list.

4.2 Implementation

4.2.1 User Interface

In addition to adding the bluetooth provider to GPM, the two other projects were also modified in order to provide a user interface that was consistent with the already existing providers. In terms of usage, the Bluetooth provider differs from the others

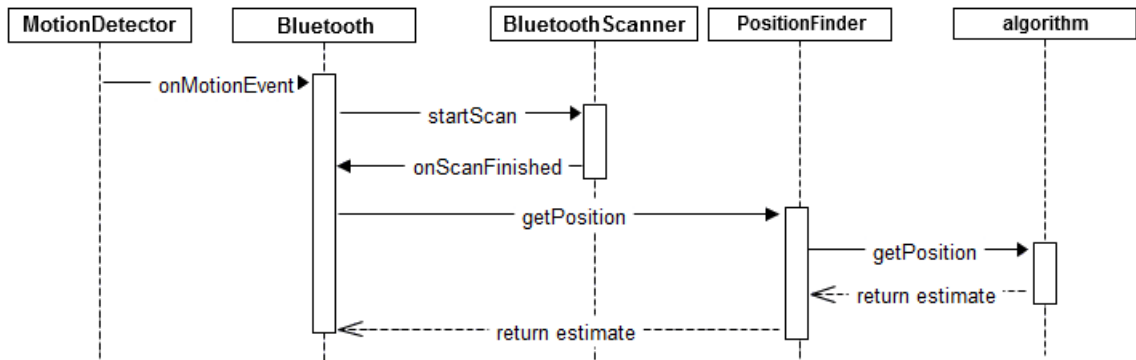


Figure 4.3: Sequence diagram for position estimation using the Bluetooth provider.

in the way that it relies on the fingerprint database that has to be created beforehand, and we therefore had to implement a separate user interface for this purpose. This interface allows the user to select fingerprint locations on a floor plan and to easily create fingerprints that are automatically associated with the selected fingerprint location.

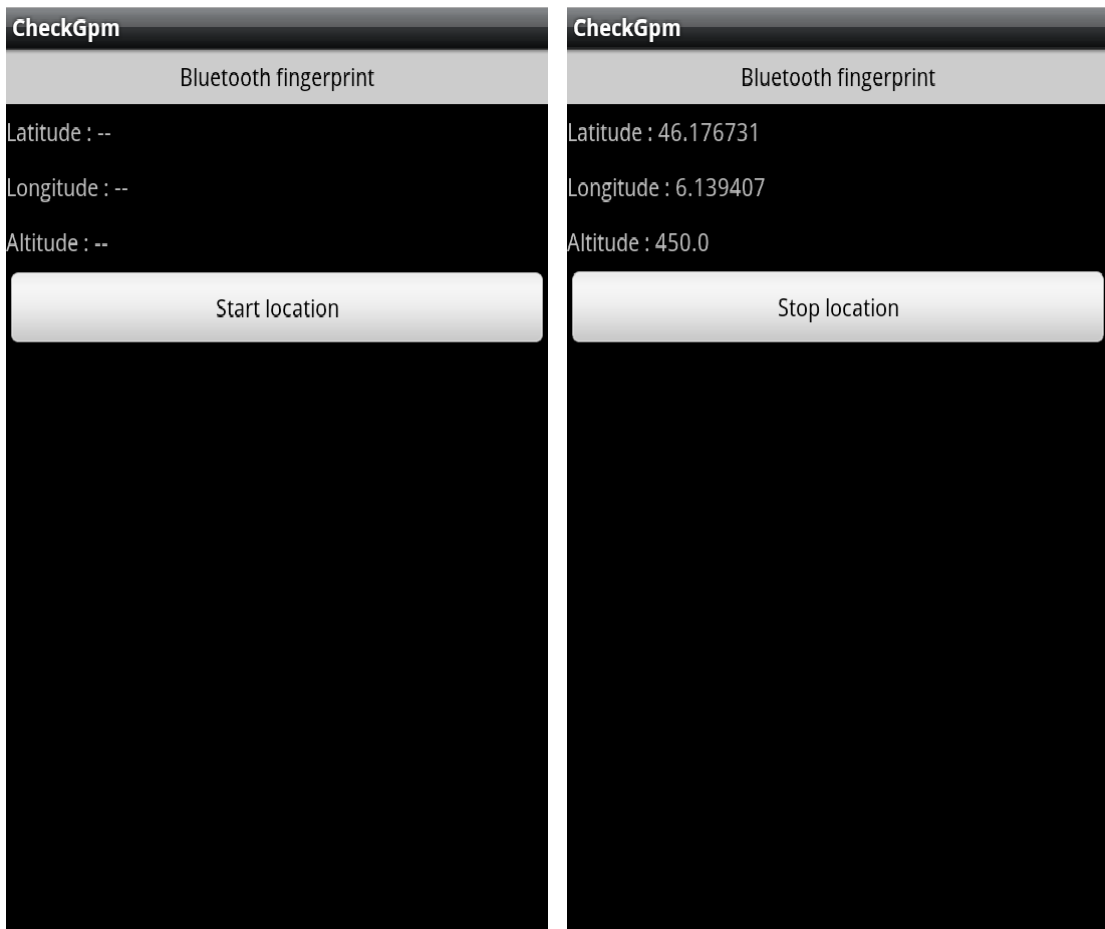
4.2.1.1 CheckGPM

CheckGPM is a small Android application intended for developers only, which is bundled with GPM. This application was updated to include a user interface for the new provider, similar to those of the other providers. The “online” screen is shown in figure 4.4. The estimated position is presented using latitude, longitude and altitude. A similar screen exists for the offline phase, where the user manually inputs the latitude, longitude and altitude of the fingerprint.

4.2.1.2 Interface for Radio Map Creation

The implementation of the user interface for radio map creation consists of three components:

- An activity for placing floor plans onto Google Maps in the form of image overlays, added to the GMA project. This activity allows the user to select an image on the sdcard to be used as an overlay, and then to move, rotate and scale the image so that it can be placed correctly on top of a building.
- The offline functionality of the Bluetooth provider in GPM
- An activity in LocMap which makes use of the above components to present the user with an interactive map on which he or she can drag and drop a pointer



(a) CheckGPM before positioning

(b) CheckGPM after positioning

Figure 4.4: The user interface of the Bluetooth provider in CheckGPM.

onto a floor plan to indicate a fingerprint location, given by the latitude and longitude provided by Google Maps. Once this has been done, the user presses a button to start an inquiry, which is then added to the radio map.

The four steps needed in order to use the positioning system are illustrated by figure 4.5, and includes the following:

1. Add a floor plan to the map
2. Select the beacons which are to be used in the system.
3. Create fingerprints on top of the floor plan
4. Obtain positioning estimates (indicated by the yellow dot).

4.2.2 Database

The radio map consists of a simple database containing three tables, as shown in the entity relationship diagram in figure 4.6.

bt_beacon contains the list of beacons that are being used by the Bluetooth provider.

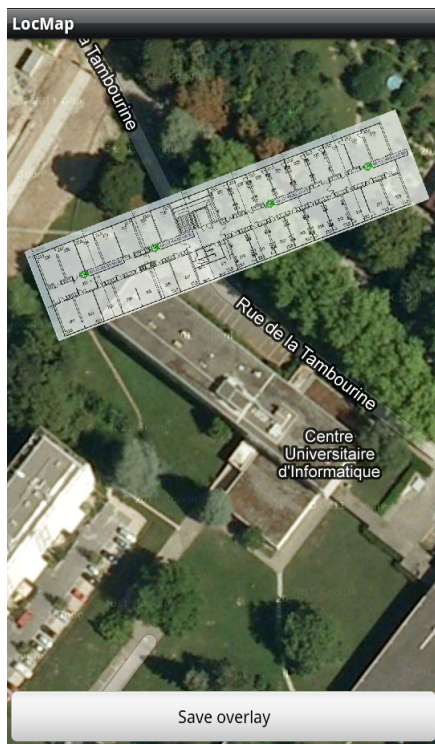
The address alone would be sufficient to identify the beacon, but the name is also stored to facilitate a presentation in a human-readable format.

bt_measurement stores the inquiries. Each row represents an individual inquiry, which is accompanied by its position in the form of the latitude, longitude and altitude (the altitude is currently not used).

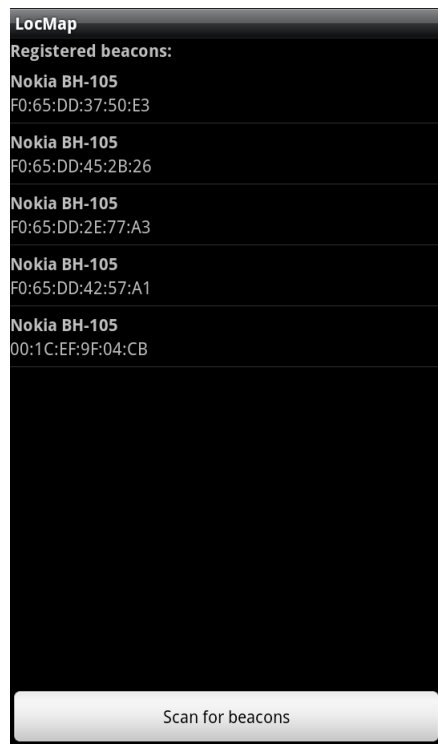
bt_beacon_reading contains the readings of individual beacons. Each row stores the address, name and RSSI of the beacon, as well as the `bt_measurement_id` which connects it to an inquiry. In other words, an inquiry consists of a set of `bt_beacon_readings`.

4.2.3 Improvements

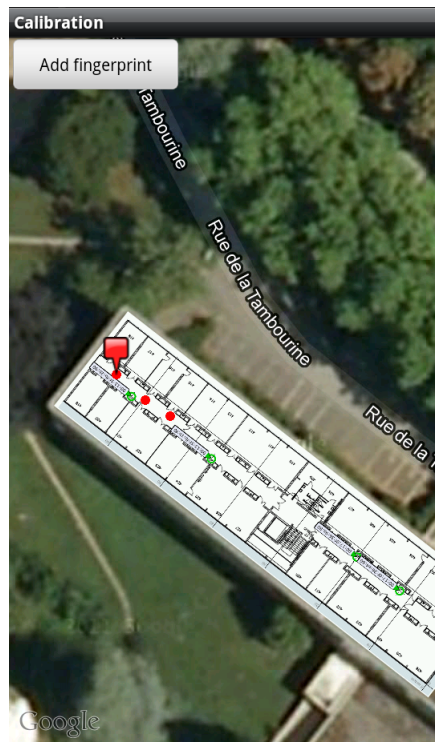
During the implementation phase of the project, certain measures were taken in order to improve the performance of the system. These are described in more detail here.



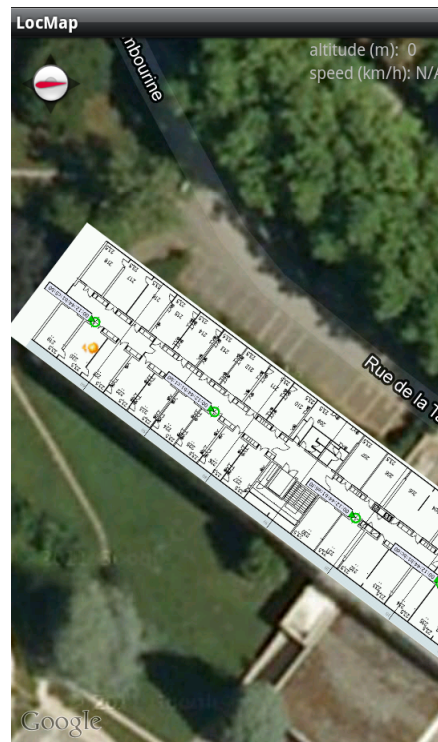
(a) Adding a floor plan



(b) The list of Bluetooth beacons



(c) Adding fingerprints



(d) Positioning

Figure 4.5: User interface for radio map creation and positioning.

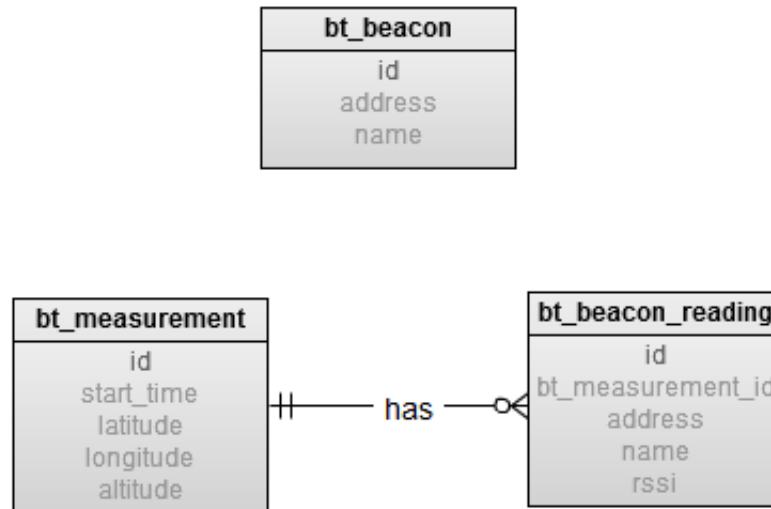


Figure 4.6: Entity relationship diagram of the GPM database.

4.2.3.1 Motion detection

Although it is difficult to measure how much battery an application uses, it is still important to make sure that it does not drain an unreasonable amount of power as this lowers the user’s satisfaction with the application.

Continuously performing Bluetooth inquiries is bad for the battery lifetime, and to remedy this situation to a certain degree, we implemented a motion detector which stops the inquiries as long as the mobile phone is not moving. We work under the assumption that when the mobile phone is not moving, the user is not moving either. This was done by taking advantage of Android’s `SensorEventListener` interface, which gives access to the build-in accelerometer. By applying a low-pass filter and removing the effect of gravity from the accelerometer’s output, we were able to detect whether the mobile phone was moving or not and to stop and start inquiries accordingly.

4.2.3.2 Shortening the Bluetooth Inquiry Time

As already mentioned, the disadvantage of the Bluetooth technology is the large delay during the discovery phase of the protocol. As pointed out in the state of the art, Hossain et al. [15] reduced this delay considerably (to approximately 4 seconds) and still managed to discover nearly 100% of the beacons. We therefore stop the inquiry procedure after a certain amount of time in order to have a more responsive system. We use an inquiry time of 6 seconds, a limit that was established empirically by

conducting several tests inside the Computer Center of the University of Geneva, using an HTC Sensation mobile phone. We were not able to obtain the same result as [15] and therefore chose to use 6 seconds.

Chapter 5

Evaluation

This chapter presents the methods used to evaluate the performance of the implemented system, and discusses the obtained results.

5.1 Scenarios

In order to properly evaluate the system, we first developed a series of scenarios. These scenarios describe how a user interacts with the system in different environments, and are used in order to derive the actual test cases. The scenarios should therefore cover all possible situations in which the system could be used. This does not mean that we create a scenario for each possible environment but rather that we try to find environments that serves as generic representatives of a group of similar environments. For each scenario that is described in the following sections, we also provide a description of the environment that it represents as well as a list of similar environments to which the scenario also applies.

5.1.1 Hospital

John goes to the hospital for a medical examination. He is not familiar with the hospital, which is a large multistory building with numerous departments. He presents himself at the reception and is given instructions on where to find his doctor. In spite of the instructions, John decides to use the map of the hospital that is available on his mobile phone to help him find the right place because the instructions are difficult to remember. When he opens the map on his phone, it shows him on which floor and in which room he is. He is also able to localize his destination on the map. With this information in mind, he starts walking quickly towards the destination while keeping an eye on the map to ensure that he is on the right way. His position is updated as

he walks, which makes it easy to see if he makes any mistakes. The position is not 100% accurate, but is always inside the correct room.

Mary works in the same hospital as a doctor. An emergency situation occurs in one of the patient rooms and her assistance is needed quickly. Because the hospital is large, she does not know the exact location of every room and she consults the building's map on her mobile phone to find out where it is. She starts running towards the room. As she runs, her position on the map is updated every few seconds, and although the position is not entirely correct, it at least indicates the correct floor and the general area of the building in which she is. When she reaches the patient room and slows down, the correctness of the position increases and indicates the correct room.

Description of a hospital: Hospitals generally have several floors which consist of rooms and corridors. Most of the rooms are patient rooms and doctors offices, hence much smaller than the kind of open areas one finds in e.g. an airport. The rooms have doors, which can be both open and closed. The rooms are furnished, either with desks, chairs and cabinets in the case of an office, or beds, nightstands, chairs, closets and possibly medical equipment in the case of a patient room.

Hospitals are generally not crowded. Patients and employees may remain at the same location for longer periods of time, but they will also walk and sometimes run, in the case of an emergency, for instance.

Similar situations:

- Universities
- Office buildings
- Retirement homes
- Underground passages such as mines and catacombs

5.1.2 Museum

Mary visits an art museum during her weekend trip to Geneva. The museum has paintings made by one of her favorite painters and she is eager to be able to see them. The museum is quite large so Mary checks the museum's map on her mobile phone to find out where the paintings are. The map shows her that they are on the other side of the building and she decides to explore the rest of the museum on her way there.

Mary walks leisurely in between the various exhibited items, occasionally stopping to look more closely at something or to read information about an item. She slowly makes her way from one room to another. The museum is quiet and apart from the occasional groups of tourists receiving guided tours, she encounters few other visitors. After a while she again remembers the paintings she came to see. She no longer knows where in the museum she is, so she takes up her mobile phone, which also displays her position on the map. She starts walking towards the paintings. Her position on the map is updated every few seconds and she can easily see that she is going in the right direction.

When she reaches the destination, she sees that a group of tourists are there and the area is quite crowded. She has to wait a bit in order to get close enough to the paintings. After having admired the paintings, she decides to leave the museum and again checks the map in order to find the direction to the exit. Guided by the automatically updated position, she finds the exit without any problems.

Description of a museum: Museums are made up of rooms of varying sizes, which generally do not have doors in between them. There are usually no corridors. The rooms generally do not have furniture, but instead have the exposed items spread out over the surface, hence contain a large number of obstacles. The visitors walk slowly and occasionally remain still in front of items. The museum may be anything between crowded and almost empty.

Similar situations:

- Visiting an art gallery
- Walking in a shopping center
- Visiting a library

5.1.3 Airport

John's flight departs from Geneva airport, an airport he has never been to before. He enters the large departure hall and looks around for his airline's check-in counter. The area is very crowded but he finally finds it after walking around for a few minutes, dodging other people who hurry past him. He lines up at the end of the long queue and waits for his turn. He finally has his luggage checked in and is given his boarding pass.

He then needs to go to the security check but he is not quite sure where it is so he takes out his mobile phone to have a look at the map of the building. The map indicates his approximate current position inside the departure hall, and using this information he figures out which direction to go. He starts walking towards the security check area while occasionally checking his position on the map to make sure that he is still headed in the right direction.

Once through the security check, he realizes that he is starting to be late and needs to hurry to the gate. He looks up his gate number and departure time on the large monitor on the wall and decides that he better run in order to make it there on time. Again, he checks his mobile phone this time to see where his gate is located. While the area after the security check is less crowded than the departure hall, there are still too many people for him to be able to actually run, and he starts walking quickly. He keeps his eyes on the mobile phone to make sure that he does not make any wrong turns but he notices that his position on the map is less accurate yet still sufficient to give him an idea of where he is. He reaches a long corridor, which is almost empty, and he takes off running. The next time he checks his mobile phone he realizes that his position on the map is wrong, and stops. A few seconds later, the position has been corrected, and again indicates the right place. This assures him that he is not lost, and he starts running again. He finally arrives at the gate as the other passengers are boarding.

Description of an airport: Airports have large halls, shopping areas and long corridors. The halls may be hundred meters long or more, have a large distance between the floor and the ceiling, and are furnished with benches and may have counters for check-in, information etc. The various shops in the shopping area have walls and ceilings but do not have doors. They vary in size, from small kiosks to large department stores. The corridors can be very long, in the range of hundred meters or more. The number of people in a particular area of an airport can range from very crowded to almost empty, and people will be remaining still, walking and occasionally running.

Similar situations:

- Inside a train station
- Inside multi-storey car parks
- Warehouses
- Factories

5.2 Preliminary Test

The environments in the scenarios described above varies mainly by the size of the rooms. We therefore conducted a test to see if the size of a room has any effect on the variability of the RSSI. If this was the case, the experiments would have to be carried out in rooms of various sizes in order to evaluate all of the scenarios. Note that we do not care about differences in the RSSI itself, since the actual value is of no importance. The only thing that matters is to be able to reproduce similar signal strengths on the same locations, that is, a small standard deviation.

We selected three different places to conduct the test:

- A 6×4 m office
- A larger, open area on the ground floor of approximately 20×15 m
- Outdoors

For each of these cases, we measured the RSSI emitted by a Nokia 5500 mobile phone 100 times, at a distance of two meters. We then calculated the mean RSSI and the standard deviation for each case, as shown in figure 5.1. Table 5.1 presents the mean RSSI and the standard deviation.

	Office	Ground floor	Outdoors
Mean RSSI	-57.05	-45.59	-74.27
Standard deviation	1.7928	3.8831	2.4694

Table 5.1: The mean RSSI and standard deviation obtained in the preliminary test.

We can see that the standard deviation stays small for all three cases and only varies by 2.09 by comparing the largest and the smallest value. Taking into account that the RSSI can vary by more than 10 simply due to the direction of the user with regard to the beacon, this is a small variation. From this we can draw the conclusion that the room size has little or no effect on the signal variation and that it is not necessary to do separate tests of the final system for office environments, open spaces and halls.

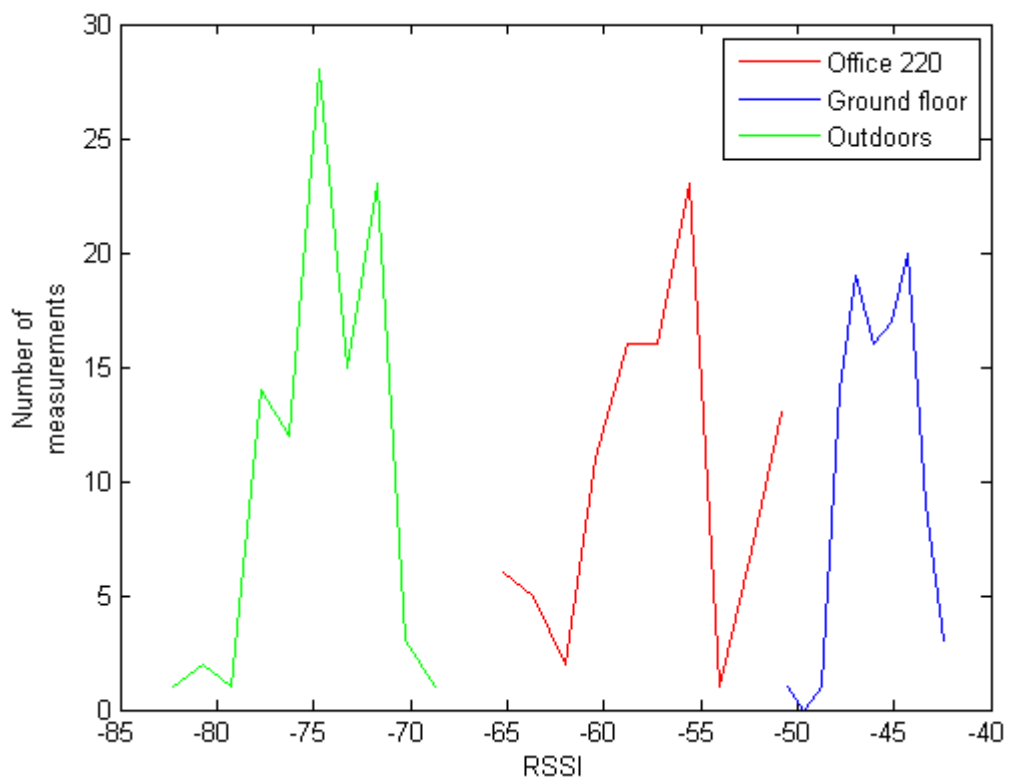


Figure 5.1: The results of the preliminary test to evaluate signal variation in rooms of varying sizes.



Figure 5.2: Nokia BH-105 Bluetooth Headset.

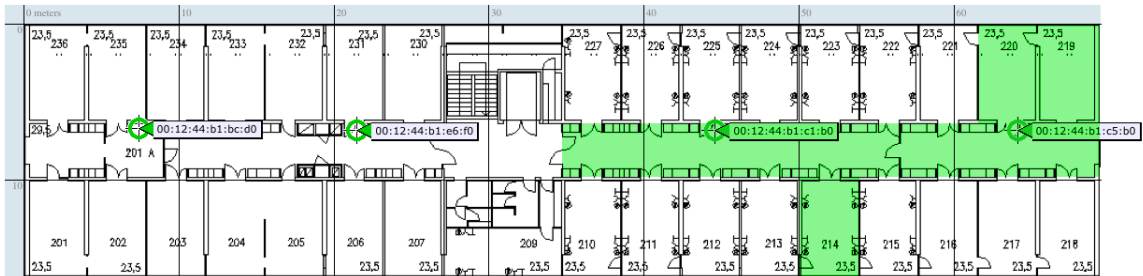


Figure 5.3: The first floor of University of Geneva's Computer Center.

5.3 Environment

The experiments were carried out using a set of Nokia BH-105 Bluetooth Headsets as beacons and an HTC Sensation mobile phone for measuring the Bluetooth signals. LocMap was installed on the HTC and used in order to create the radio map.

We chose to conduct the experiments on the first floor of University of Geneva's Computer Center. More specifically, we used offices 214, 219 and 220 as well as the corridor on the right side of the building, as shown in green color in figure 5.3. The offices are all 6×4 m large, whereas the corridor is 35 m long and 2.3 m wide. Office 219 and the corridor are shown in figure 5.4.

5.4 Experiments

This section describes the various experiments that were conducted in order to evaluate the system performance.



Figure 5.4: One of the offices (left) and the corridor (right) used in the experiments.

5.4.1 Precision and Accuracy of Positioning Algorithms

In this experiment we evaluated the precision and accuracy of the system for each of the positioning algorithms described earlier. In this report we define accuracy and precision the following way, based on the definition in [17]

Accuracy The accuracy of the horizontal positioning, in meters. It is the mean distance error between the real position and the estimated one.

Precision Three discrete values from a cumulative probability functions are used to represent the precision, namely 50, 80 et 95. A system that declares a precision of (10, 15, 30) means that there is a location precision of 50% within 10 m, 80% within 15 m and 95% within 30 m.

The first step consisted in creating a radio map of the area. Measurement locations were chosen roughly every two meters, as shown in figure 5.5, excluding the places that were covered by tables or desks. The measurements were taken by a person who was holding the mobile phone at waist height, to simulate real usage. Because Bluetooth signals are absorbed by the body, the direction of the person with regard to the beacon can greatly influence the signal. We therefore measured the signals in eight different directions, taking 1 sample in each direction. Overall, measurements were taken on 21 different locations, giving a total of $21 \times 8 = 168$ fingerprints.

A separate software was used for the calculation of the accuracy and the precision. The calculation was done using cross validation on the radio map. Since the data set (the radio map) is quite small, leave-one-out cross validation was chosen instead of the more commonly used k-fold cross validation. Leave-one-out, as the name suggests, uses a single data sample as the validation data, and the rest of the set as the training data. This is repeated for every single sample, hence it is computationally heavy but gives a more thorough validation than the often used 10-fold validation.

The positioning system uses latitude and longitude, expressed as decimal degrees with an accuracy of 6 decimal places, to denote locations. In order to calculate the distance between two points, in this case the distance between the location of the validation sample and the locations of each of the training samples, the distance had to be converted to the metric system. The accuracy of such a conversion depends both on how many decimals that are being used, and on the distance from equator¹. Due to the curvature of the Earth, the accuracy of the longitude increases with the distance

¹See http://en.wikipedia.org/wiki/Decimal_degrees for more details.

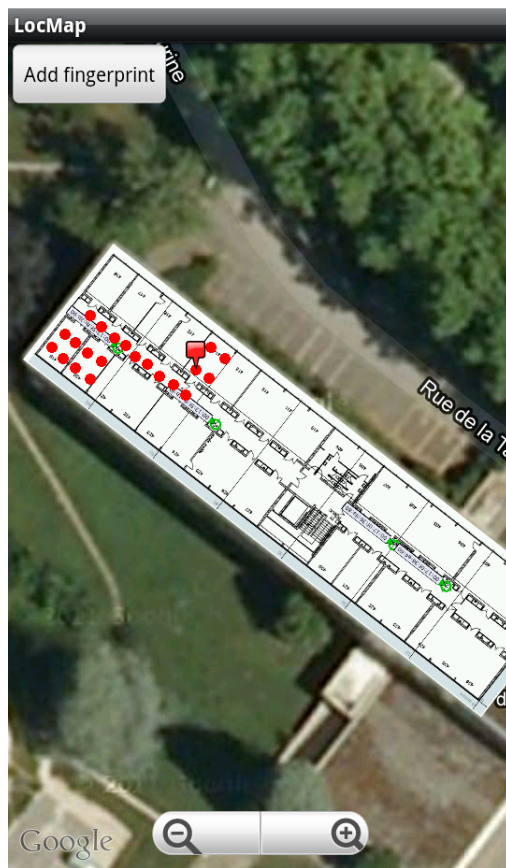


Figure 5.5: The locations where measurements were taken during the creation of the radio map (screenshot from LocMap).

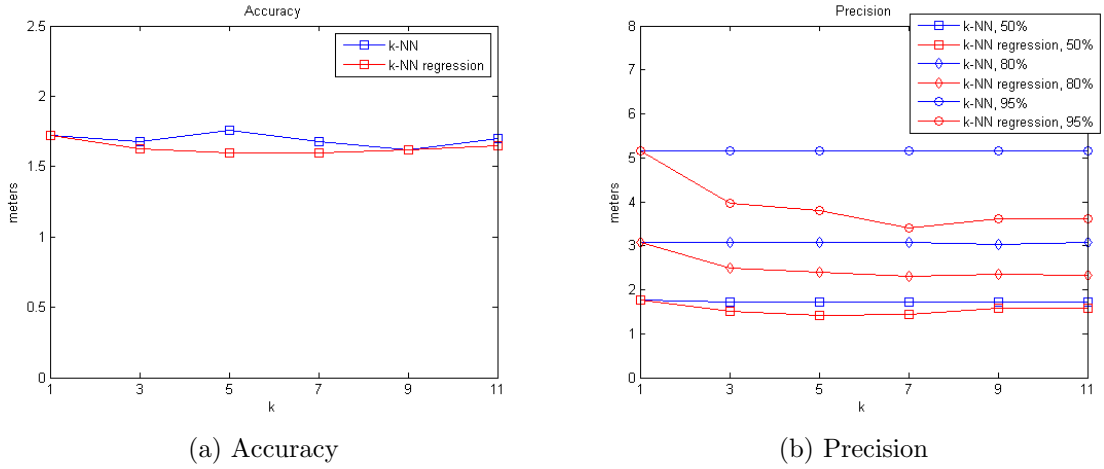


Figure 5.6: The accuracy (a) and precision (b) of k-NN and k-NN regression for different values of k .

from equator, whereas the accuracy of the latitude remains similar. At equator, a coordinate with an accuracy of 6 decimal places is accurate to 0.111 m.

The distances were converted using the *haversine* formula, which gives the distance between two points on a sphere defined by their latitude and longitude:

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat_1)\cos(lat_2)\sin^2\left(\frac{\Delta long}{2}\right)$$

$$d = R2atan2(\sqrt{a}, \sqrt{(1-a)})$$

where R is the radius of the Earth (which has a mean radius of 6371 km), and d is the distance between the two points.

5.4.1.1 Results

The results of k-NN and k-NN regression for different values of k are shown in figure 5.6. Figure 5.6a plots the accuracy, in meters, against different values of k , and we can see that the accuracy does not vary much with regard to k . Yet, the best accuracy was obtained using $k = 9$ (1.62 m) for k-NN and $k = 7$ (1.60 m) for k-NN regression.

Figure 5.6b shows the precision for 50%, 80% and 95% for k-NN and k-NN regression. The regression algorithm obtained better results in all cases, in particular in the case of 95%. This can easily be explained by the fact that it computes an average of the k points, hence the predicted position is not limited to the actual measurement points as is the case for k-NN. Overall, the best precision is obtained with $k = 7$ for regression and $k = 9$ for k-NN.

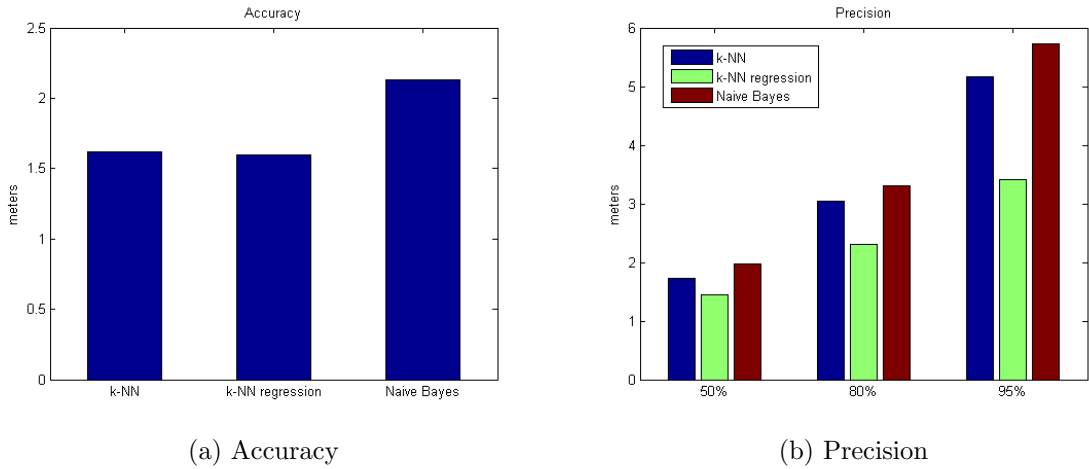


Figure 5.7: The accuracy (a) and precision (b) of k-NN (for $k = 9$), k-NN regression (for $k = 7$) and Naïve Bayes' Classifier.

Next, we compare the performance all three algorithms. Figure 5.7b shows the precision obtained for k-NN with $k = 9$, k-NN regression with $k = 7$ and the Naïve Bayes' Classifier, and figure 5.7b shows the precision of the algorithms. The actual numbers are presented in table 5.2. First of all, Naïve Bayes' Classifier performed

	Accuracy (m)	Precision (m)		
		50%	80%	95%
k-NN	1.62	1.72	3.04	5.16
k-NN regression	1.60	1.42	2.40	3.79
Naïve Bayes	2.13	1.97	3.31	5.72

Table 5.2: The accuracy and precision obtained for k-NN, k-NN regression and Naïve Bayes' Classifier.

worse than the others both on accuracy and on precision, although the precision is close to that of k-NN. Again, this can be explained by the way a classifier is limited to the actual classes, or positions in our case, and cannot interpolate. Not surprisingly then, k-NN regression has the best precision out of the three with a precision within 1.42 m 50% of the time, within 2.40 m 80% of the time and within 3.79 m 95% of the time. k-NN and k-NN regression has similar accuracies of about 1.6 m, a difference of about 0.5 m to that of Naïve Bayes.

A particularity worth mentioning is the fact that when a classifier predicts the correct position, the error is 0 m, which significantly lowers the average error.

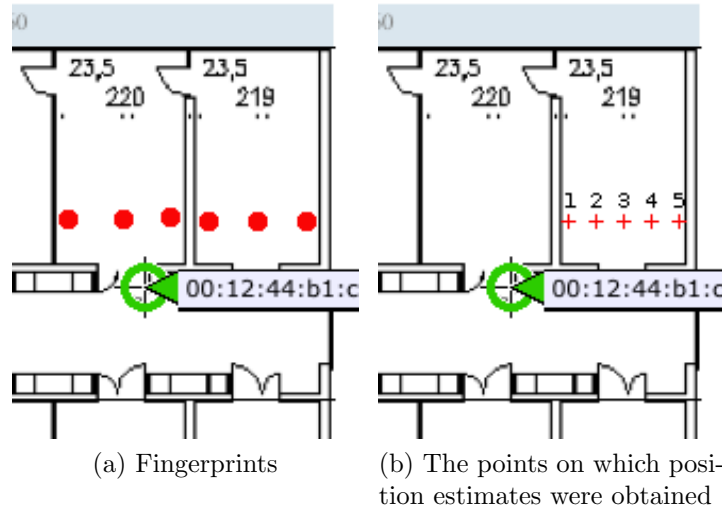


Figure 5.8: Experimental set-up for evaluating the room-level accuracy.

It is important to note that this is an optimal scenario. The experiment site remained static during the whole experiment: there were no other people present, furniture were not moved, doors were not opened or closed. In a real setting where the environment changes we expect to obtain less accurate results.

5.4.2 Room-level precision

In addition to knowing the accuracy and precision of the system, we were also interested in knowing to which degree the system can predict the correct room. This is particularly relevant for the hospital scenario, in which a very precise positioning might not be necessary but the ability to find the correct room in an emergency situation is of utter importance.

The experiment consisted of two offices, in which fingerprints were created on six different points, three in each office, as shown in figure 5.8a. Then, positions were estimated at the five points marked in figure 5.8b. The points are slightly less than one meter apart, and point 1 and 5 are less than 0.5 m from the wall.

At each of the five locations we did 20 measurements and counted the number of times that the predicted position was inside the correct room. This particular set-up was chosen in order to see if there are differences in the room-level accuracy when the device is close to the wall compared to when it is in the middle of the room.

5.4.2.1 Results

The results are presented in table 5.3. Here, we can see that the error rate generally

Position	1	2	3	4	5
Correct estimates	75%	95%	95%	100%	95%

Table 5.3: The percentage of position estimates inside the correct room.

stays very low but increases to 25% when the device is close to the wall. This happens because the signals coming from the neighboring room are not diminished sufficiently by the wall in order for the system to be able to completely distinguish between the two fingerprints on each side of the wall. To remedy this situation one could envisage the system to take the current position of the device into account and forbid moves that are physically impossible. Such a solution is proposed in [6] where the author uses a *location graph* to modelize the way that points of interest are connected to each other in an indoor environment.

5.4.3 Performance When Walking

The goal of this experiment was to evaluate system performance when the device is moving. A drawback of Bluetooth is the large delay caused by the inquiry procedure which prevents a system based on this technology to provide a continuously updated position. This is generally not a problem when the device tends to remain at the same place for long periods of time but could hamper the system if the device is expected to move around a lot.

We used the corridor shown in figure 5.3 for this purpose. Fingerprints were created approximately every two meters, each consisting of three measurements in the direction towards the middle of the building, as shown in figure 5.9

The experiment was performed by walking down the corridor in the same direction as the fingerprint measurements, while holding the mobile phone. This was repeated using various walking speeds.

5.4.3.1 Results

We observed the following:

- When walking very slowly, the correctness of the position estimation was similar to when standing still but tended to lag behind.
- As the walking speed increases, the correctness decreases. When doing a fast walk, the estimations were no longer reliable and tended to “jump” between different areas of the corridor.

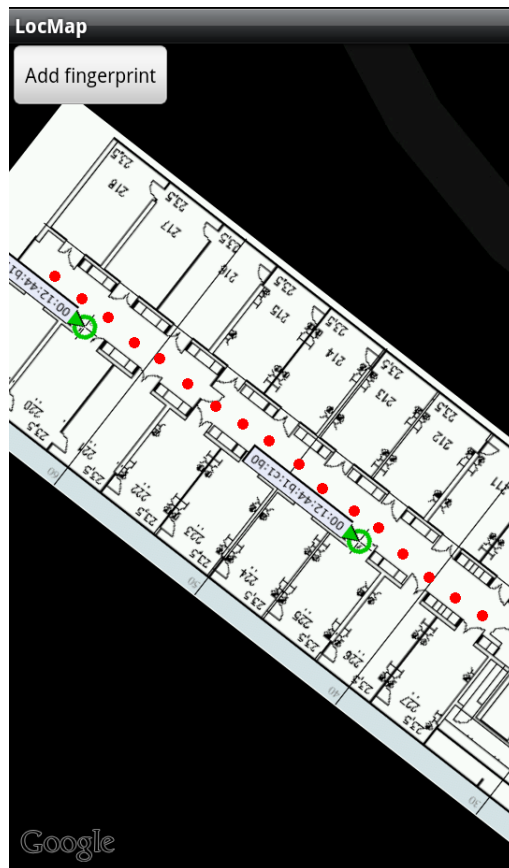


Figure 5.9: LocMap screenshot: fingerprint locations used for evaluating system performance when walking.

- When stopping, the correctness is restored as soon as the succeeding inquiry has finished, as was expected.

This is due to the latency described earlier. When the user moves too fast, the signal strengths registered during the inquiry will be different from those that are stored in the radio map because the distance between the user and the various beacons, and hence the signal strengths, change during the scan itself. This causes the resulting measurement to be a “mixture” of several fingerprints and will have a low similarity with the fingerprints in the radio map.

One can speculate that the solution proposed in the previous section could somewhat improve the accuracy when the user is moving since it would suppress estimates that are physically impossible to have been reached from the previous position of the user. Another option could be to create a radio map containing fingerprints created while actually moving. To our knowledge, there is currently no research which explores such a solution. We can conclude from this that the system is unable to provide sufficiently correct position estimates while the user is moving at speeds faster than slow walking.

5.5 Comparison with other indoor positioning systems

Liu et al. [17] presents an overview of various indoor positioning systems and their performance. Comparing our results to those presented there, we can conclude the following:

- Positioning systems based on WLAN technology and the received signal strength generally obtains an accuracy of 2 to 5 meters, although there are a couple of noteworthy exceptions, such as Ekahau, which has an accuracy of 1 meter. Our results are similar to or better than the majority of those systems. Due to the high stability of the Bluetooth RSSI compared to the WLAN RSS, it is simpler to implement systems based on signal comparisons when using Bluetooth. This is an argument for choosing this particular technology.
- The precision of the WLAN systems ranges from 1.5 m to 5 m 50% of the time. This is comparable to our result of about 1.5 m 50% of the time, taking into account that the precision in an actual deployment of the system would likely be lower, as mentioned in the previous sections.

- The only Bluetooth system presented in the overview is TOPAZ, which combines Bluetooth and infrared. TOPAZ has an accuracy of 2 m and a precision of 2 m 95% of the time. The accuracy is similar to ours, whereas the precision is vastly better. The system's robustness is, however, described as "poor", indicating that such a high precision is not always possible to attain.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this project, an indoor positioning system based on Bluetooth RSSI was implemented and its performance evaluated. This report presents an overview of the current techniques being used in such systems and discusses in more details the choices that were made with respect to positioning techniques and Bluetooth signals. The evaluation was done by carrying out several experiments in order to establish the accuracy and precision of the implemented system under various conditions. From the obtained results we can draw the following conclusions:

- The implemented system has a performance that is similar to that of systems based on WLAN RSS as well as other Bluetooth based systems. The Bluetooth RSSI has the advantage of being very stable, thus more appropriate for the fingerprinting technique since filtering algorithms are not necessary, as is the case with WLAN RSS. Bluetooth is also widely available in mobile phones, computers and headsets and is a low-cost technology. The disadvantage is that if one does not wish to use a system comprised of Bluetooth dongles attached to computers, there are at the time of writing no good commercially available options for Bluetooth beacons. Bluetooth beacons intended for indoor positioning purposes simply does not exist. The tests and evaluations of the implemented system were conducted using mobile phones and headsets, which has the disadvantage of quickly running out of battery, and some also have a very short discovery phase, meaning that they often have to be restarted. Neither of these could be used in an actual deployment of the system.
- Although the obtained accuracy and precision does not differ much from what others have obtained, our system has the added benefit of being part of a larger

API that allows to easily change between different providers. The addition of the Bluetooth provider is an improvement of this API since it makes it more complete and gives the user an additional option for position estimation. This integration was the main goal of the project, and it has been successfully achieved.

- The system has a fairly good precision with 50% probability of being within ~ 1.5 m of the actual position when the mobile phone on which the system is running does not move. When the user is moving, the precision decreases drastically as soon as he or she is moving faster than a slow walk. During fast walking, the system no longer gives reliable results. We therefore conclude that this system is not suitable when it is a requirement to have accurate position estimates when the user is moving.
- As there is no deterministic relationship between the RSSI and the distance to the beacon, a fingerprinting approach was chosen. In order to evaluate which positioning algorithm was the most suitable, we chose to evaluate three algorithms that were chosen on the basis that they are frequently used in positioning systems: Naïve Bayes' Classifier, k-NN and a variation on k-NN that uses regression. Our results shows that the k-NN regression algorithm obtained the best accuracy and precision of the three, using $k = 7$. One must note that the value of k should be evaluated on a per-radio map basis as it does not necessary apply to another radio map created in a different setting.
- Taking into account the disadvantages mentioned above, one needs to carefully consider whether Bluetooth fulfills ones requirements before implementing a positioning system based on it. If some of these disadvantages are not acceptable, such as not having accurate positioning while moving, it might be worth while to also consider other options, such as WLAN.

6.2 Future work

This section presents the issues which need to be addressed in order to improve system performance.

6.2.1 Evaluate beacon placement and fingerprint locations

The optimal placement of beacons needs to be decided on a per-case basis due to the large number of variables to consider, some of which could possibly diminish performance, such as where it is physically possible to install a beacon with regard to cabling or access to power outlets etc. Yet, it should be possible to make certain general assumptions regarding

- how far apart should beacons be placed in order to get sufficient signal coverage.
- the height of the beacon.
- how close should fingerprints locations be to each other.
- how many fingerprints are needed on each fingerprint location.

In order to evaluate the points listed above we should experiments with different configurations of both beacons and fingerprint locations. Little research has been published on this specific issue, but a solution is proposed by Chawathe [5], which is based on an algorithm for creating a beacon hypergraph and relying solely on beacon visibility.

6.2.2 Scalability

The scalability of the system should be considered since a large environment could have a detrimental effect on the positioning algorithm used. The k-NN algorithm for instance, considers every single fingerprint at runtime, which could lead to delays in the position estimation if the number of fingerprints is large.

6.2.3 Improve accuracy

During the system evaluation it was observed that the position estimates would occasionally “jump” to places which were physically impossible for the user to reach, given his or hers current position and the duration of the inquiry. For instance, when the user was close to a wall, the estimated position would occasionally be on the other side of the wall. Such cases could be avoided by making the system take the physical environment into account and exclude those impossible estimates. This would involve creating a model of the building which contains the possible paths to get from one point to another.

Such a solution could also help reduce the large delay incurred by the inquiry procedure since the set of possible positions would be reduced to only those that are close to the user, and one might be able to predict the correct position before the scan has completed. This could also improve the accuracy of the system while the user is moving. One such solution is proposed in [6].

References

- [1] Marco Altini, Davide Brunelli, Elisabetta Farella, and Luca Benini. Bluetooth indoor localization with multiple neural networks. In *Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on*, pages 295–300, may 2010.
- [2] G. Anastasi, R. Bandelloni, M. Conti, F. Delmastro, E. Gregori, and G. Mainetto. Experimenting an indoor bluetooth-based positioning service. In *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*, pages 480–483, may 2003.
- [3] M.Z. Barahim, M.R. Doomun, and N. Joomun. Low-cost bluetooth mobile positioning for location-based application. In *Internet, 2007. ICI 2007. 3rd IEEE/IFIP International Conference in Central Asia on*, pages 1–4, sept. 2007.
- [4] Mortaza S. Bargh and Robert de Groote. Indoor localization based on response rate of bluetooth inquiries. In *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments, MELT '08*, pages 49–54, New York, NY, USA, 2008. ACM.
- [5] S.S. Chawathe. Beacon placement for indoor localization using bluetooth. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 980–985, oct. 2008.
- [6] S.S. Chawathe. Low-latency indoor localization using bluetooth beacons. In *Intelligent Transportation Systems, 2009. ITSC '09. 12th International IEEE Conference on*, pages 1–7, oct. 2009.
- [7] Kenneth C. Cheung, Stephen S. Intille, , and Kent Larson. An inexpensive bluetooth-based indoor positioning hack. In *Proceedings of UbiComp 2006 Extended Abstracts*. Springer-Verlag, 2006.

- [8] Javier J. M. Diaz, Rodrigo de A. Maues, Rodrigo B. Soares, Eduardo F. Nakamura, and Carlos M. S. Figueiredo. Bluepass: An indoor bluetooth-based localization system for mobile applications. In *Computers and Communications (ISCC), 2010 IEEE Symposium on*, pages 778–783, june 2010.
- [9] D. C. Dimitrova, U. Burgi, G. Martins Dias, T. Braun, and T. Staub. Inquiry-based bluetooth parameters for indoor localisation - an experimental study. In *5th ERCIM workshop on e-Mobility*, pages 13–21, june 2011.
- [10] Silke Feldmann, Kyandoghene Kyamakya, Ana Zapater, and Zighuo Lue. An indoor bluetooth-based positioning system: Concept, implementation and experimental evaluation. In *International Conference on Wireless Networks*, pages 109–113, 2003.
- [11] Tiago Fernandes. Indoor localization using bluetooth. In *6th Doctoral Symposium in Informatics Engineering*, pages 480–483, 2011.
- [12] Christoph Fuchs, Nils Aschenbruck, Peter Martini, and Monika Wieneke. Indoor tracking for mission critical scenarios: A survey. *Pervasive and Mobile Computing*, 7(1):1–15, 2011.
- [13] Yanying Gu, A. Lo, and I. Niemegeers. A survey of indoor positioning systems for wireless personal networks. *Communications Surveys Tutorials, IEEE*, 11(1):13–32, quarter 2009.
- [14] Simon Hay and Robert Harle. Bluetooth tracking without discoverability. In Tanzeem Choudhury, Aaron Quigley, Thomas Strang, and Koji Sugiuma, editors, *Location and Context Awareness*, volume 5561 of *Lecture Notes in Computer Science*, pages 120–137. Springer Berlin / Heidelberg, 2009. 10.1007/978-3-642-01721-6_8.
- [15] A.K.M. Hossain and Wee-Seng Soh. A comprehensive study of bluetooth signal parameters for localization. In *Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on*, pages 1–5, sept. 2007.
- [16] Mikkel Baun Kjærgaard. Indoor positioning with radio location fingerprinting. *CoRR*, abs/1004.4759, 2010.

- [17] Hui Liu, H. Darabi, P. Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1067–1080, nov. 2007.
- [18] Anil Madhavapeddy and Alastair Tse. A study of bluetooth propagation using accurate indoor location mapping. In *The Seventh International Conference on Ubiquitous Computing (UbiComp)*, pages 105–122, 2005.
- [19] O. Rashid, P. Coulton, and R. Edwards. Implementing location based information/advertising for existing mobile phone users in indoor/urban environments. In *Mobile Business, 2005. ICMB 2005. International Conference on*, pages 377–383, july 2005.
- [20] F. Subhan, H. Hasbullah, A. Rozyyev, and S.T. Bakhsh. Indoor positioning in bluetooth networks using fingerprinting and lateration approach. In *Information Science and Applications (ICISA), 2011 International Conference on*, pages 1–9, april 2011.
- [21] Teldio. *ZONITH Indoor Positioning Module white paper version 2.1a*.