# Push-to-fix indoor positioning using Bluetooth Low Energy

Reinoud Elhorst
Clare Hall


*A dissertation submitted to the University of Cambridge in partial fulfilment of the requirements for the degree of Master of Philosophy in Advanced Computer Science*

University of Cambridge
Computer Laboratory
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom

Email: re302@claude.nl

June 13, 2014

# Declaration

I Reinoud Elhorst of Clare Hall, being a candidate for the M.Phil in Advanced Computer Science, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

Total word count: 14981

**Signed**:

**Date**:

Released under a Creative Common Attribution-ShareAlike 4.0 International license
All trademarks used in this dissertation are hereby acknowledged.

# Abstract

Positioning systems that use the signal strength from a Wi-Fi signal, are being used by smartphones to position in locations where no GPS signal is available, such as indoors. Bluetooth Low Energy (BLE) is a relatively new technology, that has the potential to be used in a similar role, while being easy to deploy and having the potential to give high accuracy. Positioning though BLE can be done in a way that guards the user's privacy.

In this dissertation I show that the signal strength for BLE signals varies a lot with small changes in the environment. I detected changes of 20dB+ while rotating the phone, moving it by centimetres, or having a person move between the transmitter and the phone. I further show a busy environment where 59% of BLE packets are dropped; even in a quiet environment only 76% of packets are received.

I show that a push-to-fix positioning system based on Bluetooth Low Energy can be built using the same positioning algorithms as used for Wi-Fi: signal space distance (SSD). An orientation-aware variant (SSD-O) adjusted for BLE scores a 2.55m 95th percentile error. The positioning algorithms I introduce, *Blackout Resistant Positioning* (BRP) and an orientation aware version BRP-O, perform slightly worse, but a combination of BRP-O with SSD-O brings the 95th percentile error down to 2.40m.

If positioning is attempted after only 0.5 seconds, as opposed to 2 seconds before, the BRP and BRP-O methods outperform SSD and SSD-O, having a 95th percentile error of 4.63m and 3.64m respectively, against 5.38m and 4.73m for SSD and SSD-O.

Finally this dissertation compares some of the privacy, security and usability properties of BLE based positioning to other positioning methods. It shows that the user's privacy can be preserved in a BLE based positioning system, if implemented in the right way.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Using digital devices to find one's position has been one of the great advancements in technology in the last couple of decades, and has more recently entered the domain of personal electronics. At the end of 2013, it was reported that more than one in five persons owns a smartphone[1], and many of these smartphones include GPS (General Positioning System) receivers. In addition, many GPS receivers have been sold outside of smartphones, in standalone navigation devices, or built in to larger products, such as cars. Since in 2000 the Selective Availability of GPS ended, the accuracy has improved to within a couple of meters[2]. More technological enhancements are possible to reduce the error into the centimeter range[3]. The GPS system (and similar systems such as the EU's Galileo, Russia's GLONASS, India's IRNSS and China's Beidou-2 system, collectively known as *Global Navigation Satellite Systems* or GNSS), have become the standard for outdoor positioning.

The signals from GNSS satellites are weak, and usually unable to penetrate structures. This means that the system's use degrades considerably between high-rise buildings, and cannot be used indoors. In addition, indoor positioning is harder, because there are more variables; while outdoors one can

[1] `http://www.businessinsider.com/smartphone-and-tablet-penetration-2013-10`, accessed on 8 June 2014.

[2] `http://www.gps.gov/systems/gps/modernization/sa/`, accessed on 8 June 2014.

[3] `http://en.wikipedia.org/wiki/GPS_enhancement`, accessed on 8 June 2014.

usually assume that the positioning device is on ground level, indoors positioning needs to be done in all 3 dimensions. Finally, indoor positioning for consumer use has stricter requirements than outdoors: while outdoors an error of 10 meters can easily be spotted and corrected for by a user, indoors 10 meters can mean a different floor, or even building. Where GPS is not available, current smartphones use a combination of mobile signals from cell towers, and Wi-Fi signals, use of which was pioneered by Bahl and Padmanabhan [2000]. The choice for these technologies is at least partly because both the beacon infrastructure and the smartphone hardware to receive the signals was already in place. Positioning through Wi-Fi is accurate to tens of meters in a typical urban environment [Zandbergen, 2009], and the precision increases with the number of access points in close vicinity. Using mobile signals for positioning gets an accuracy in hundreds of meters.

## 1.1 Bluetooth Low Energy

Bluetooth Low Energy (BLE), also known as Bluetooth Smart, or sometimes incorrectly referred to as Bluetooth 4.0[4], is a new wireless connection technology that is quickly gaining in popularity. It was introduced in 2010 as part of the Bluetooth 4.0 standard. BLE is not compatible with previous versions of Bluetooth (now often referred to as Bluetooth Classic), and is not meant to replace traditional Bluetooth.

BLE uses the same 2.4GHz frequency band as Wi-Fi, Bluetooth Classic and many other radio devices. In order to be able to operate in this busy band, BLE uses 40 channels, and during a connection an effort is made to only use channels with little interference. Out of the 40 channels, channel 0 to channel 36 are used after a connection is made, and three channels (channel 37, 38 and 39) are advertising channels. These advertising channels are used for discovery, connectionless information broadcast and connection initialization.

[4]Bluetooth 4.0 is the fourth incarnation of the Bluetooth standard, which contains, among other things, Bluetooth Low Energy. The latest Bluetooth standard is 4.1; all claims in this report are valid for both versions.

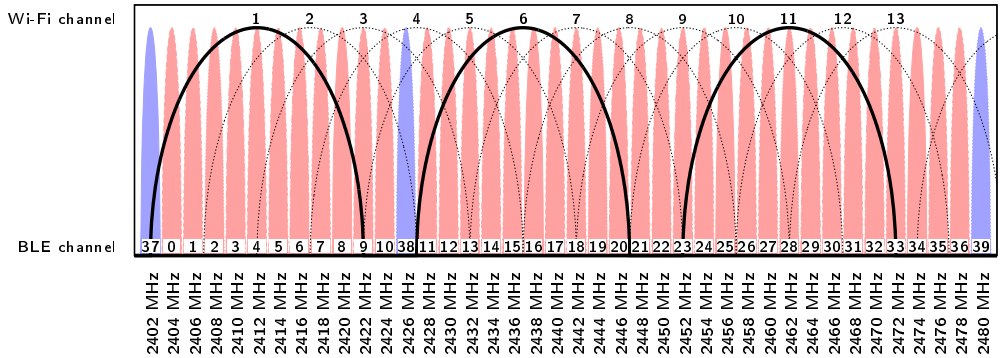



Figure 1.1: Bluetooth Low Energy advertising (blue) and data (red) channels, and Wi-Fi channels, with the much-used 1/6/11 channel combination in bold.

The three advertising channels are, as figure 1.1 shows, spaced throughout the frequency spectrum, in order to minimise the chance that all three channels suffer from interference simultaneously [Heydon, 2013].

Bluetooth Low Energy is widely expected to be available in billions of devices within a couple of years. In February 2014, all of the top 10 most sold smartphones support BLE[5]. BLE transmitters and receivers are cheap (chips such as the rfd22301, which combines an ARM M0 core and a complete BLE system, are widely available for around $15), and can run for years on small batteries[6]. BLE is designed to allow many devices to co-exist in the same area [Heydon, 2013], and Android and iOS devices can listen for, interact with and connect to BLE devices directly from the app without user intervention.

The above means that it is possible to relatively cheaply fill an area with a dense network of battery powered Bluetooth Low Energy transmitters (beacons), which can be interacted with by apps running on hardware that most people carry with them anyway. This makes BLE a plausible candidate to provide indoor positioning.

[5] `http://www.counterpointresearch.com/top-10-smartphones-in-february-2014`, accessed on 4 June 2014.

[6] `http://www.bluetooth.com/Pages/low-energy-tech-info.aspx`, accessed on 4 June 2014.

## 1.2 Positioning

*Radiolocation* is the technique of determining a device's location by use of radio waves. This technique can be used to either determine location on the basis of a single measurement, or to track a device's movement over time. The former is called *push-to-fix positioning* or *one-shot positioning* (in this report I will usually just call it *positioning*), while the latter is known as *tracking*. In this report I exclusively focus on push-to-fix positioning, however I believe that an improvement in push-to-fix positioning can also be used to improve tracking performance.

Radio-positioning can broadly be done in two ways. Either the device to be located can send a signal, which is received by stations in the environment, which determine the device's location. Alternatively, stations in the environment can send out signals, which are received by the device. Using these signals the device can determine its location. This latter form is known as *opportunistic positioning*. Both GPS and the Wi-Fi positioning system (WPS) are examples of this latter form.

Several techniques can be used to do radiolocation; I mention some of the most used. Time-Of-Arrival and Time-Difference-Of-Arrival use the fact that the radio waves travel at a limited speed to determine location. This method is used by GPS, and requires accurately timed hardware, which is most likely not present in consumer BLE chips (however in section 3.8.1 I suggest a system that makes use of this technique in a different form). Angle-Of-Arrival uses the direction from which a signal arrives to determine the location. Even though most mobile phones do not have the hardware to directly determine the direction a BLE signal comes from, it is possible that something could be inferred indirectly from the signal strength; this is being left for further research however. Positioning using Received Signal Strength (RSS; or Received Signal Strength Indication, RSSI) uses the fact that a signal has a different strength in different locations. RSS is being used in many Wi-Fi positioning systems, and both iOS and Android expose the RSS of received packets to apps. Sometimes the term path-loss is being used as well in this

context. Path-loss defines how much the signal strength has dropped between transmitter and receiver, and is defined as the difference between transmitter power and RSS. Path-loss based and RSS based positioning therefore describe the same thing in most situations.

## 1.3 Previous work

Research into indoor positioning has a long history. One of the first systems was the *active badge* system, described in Want et al. [1992], which uses badges with infra-red transmitters to determine one's location. The *active bat system* [Harter et al., 2002] and the *cricket system* [Priyantha et al., 2000] use a combination of radio and ultrasound signals to do positioning. Bahl and Padmanabhan [2000] introduced Wi-Fi positioning using the RSS from Wi-Fi access points and a fingerprint database to determine a device's location indoors. King et al. [2006] uses the compass orientation during both surveying and positioning to improve accuracy. Castro et al. [2001] further looked into Wi-Fi positioning, and Pandya et al. [2003] suggested using Bluetooth (pre-4.0, hence not BLE) to improve indoor positioning. Li et al. [2005] looked at improved methods for building the fingerprint database, and Shin et al. [2012] suggested improvements on the nearest-neighbour positioning algorithm. A completely different approach to positioning, using techniques from machine learning, can be found in Battiti et al. [2002] and [Ferris et al., 2007].

## 1.4 BLE based positioning

To my knowledge there is no published academic work on using Bluetooth Low Energy for positioning.

The Bluetooth 4.0 standard does contain a *proximity profile (PXP)*[7]. Using

[7] `https://developer.bluetooth.org/TechnologyOverview/Pages/PXP.aspx`, ac-

PXP, devices can receive alerts when they move out of range of another device; this is done using RSS.

In July 2013 Apple introduced the iBeacon technology. An iBeacon sends Bluetooth Low Energy (BLE) packets at regular intervals, and a BLE capable device (such as an iPhone) can receive them and determine its proximity to the beacon. Using this technology, iOS[8] devices can offer location-aware functionality when inside an iBeacon-region[9].

Even though both systems allow for a certain level of location awareness, this is relative to a certain device or beacon. There is no evidence that either is intended to be used to determine a location more accurately than close or far from a certain device or beacon.

A positioning system using BLE has the following requirements.

- Technical: Positioning has to be fast, with few errors. In chapter 3 I explore this, and show this is possible using BLE.

- Economical: It has to be cheap to install the infrastructure. In the case of BLE, ever more people carry receivers with them, and, as discussed before, simple beacons will not have to cost much more than $20 each. Even with a dense deployment on a beacon every 2 meters, this results in only $5 per $m^2$.

- Social: People may be worried about security and their privacy, both towards companies and towards the government, and therefore avoid certain positioning technologies. Having a technology that can be used without anyone even knowing it was used will enhance a feeling of privacy. Chapter 4 shows that only a little privacy has to be given up to work with BLE positioning, comparable to Wi-Fi positioning, if the right technology is used.

---

cessed on 4 June 2014.

[8] and Android, although the system was developed by Apple for iOS devices

[9] `https://developer.apple.com/ios7/`, accessed on 4 June 2014.

## 1.5 Material

The experiments in this report were done using CSR BLE beacons as transmitters and an iPhone 5 (section 2.3), an iPad mini (section 2.5) and an iPhone 5S (all other sections), all running stock iOS 7, as receivers. The CSR beacon is a device that was developed by CSR to service as a battery powered indoor BLE beacon. The iPhones were chosen because smartphones are the typical devices one may want to use for positioning, the maturity of the iOS BLE implementation, and the author's previous experience with iOS. In addition iOS returns the channel on which a packet is received (see section 2.2), something Android does not; this turned out to be extremely useful in some tests.

The iPad was chosen, because its standing form-factor and long battery life[10] made it more suitable for the long-term measurements in section 2.5.

Initially a Nexus 4 and Samsung S4 running Android were used as well, however after several occasions in which the devices had not logged the requested data, their use was discontinued. Anecdotal evidence suggests that the BLE stack on Android had some stability issues.

## 1.6 Contributions

In this report I look at the possibilities for using Bluetooth Low Energy for push-to-fix positioning. In chapter 2, I explore some of the properties of BLE that one has to keep in mind when positioning, such as the fact that there are three different advertising channels, the influence multi-path interference, orientation and moving objects have on the RSS, and the packet loss of BLE advertising packets. Chapter 3 shows how a positioning system

[10]Battery impact of listening for BLE packets was not explicitly tested in these experiments. Since an app has to stay in the foreground to receive BLE packets at a high rate, the screen had to be kept on during scanning, which will have negatively impacted the battery; still multiple hours of capturing BLE packets was possible on the iPhone 5 without recharging.

based on BLE beacons works in the Cambridge University Computer Lab. As main contribution I introduce *Blackout Resistant Positioning* in section 3.3.4, a new algorithm to do positioning, that takes specific Bluetooth Low Energy properties into account, and which is shown to outperform Single Space Distance (SSD), an algorithm frequently used in signal strength based positioning, in many cases. Secondary contributions lie in the exploration of positioning algorithms used for Wi-Fi positioning by Bahl and Padmanabhan [2000] and King et al. [2006] and their adaptation to BLE positioning. An additional contribution is a discussion in chapter 4 on how a positioning system based on BLE performs on privacy and security, compared to other positioning methods.

## 1.7 Research questions

I focus on the following questions in this report:

- What are the radio propagation properties one has to take into account when trying to build a push-to-fix Received Signal Strength (RSS) based positioning system based on Bluetooth Low Energy?

- Can we use the positioning methods that were developed for Wi-Fi positioning for BLE positioning? Can I find an RSS-based push-to-fix positioning algorithm that works better than these, by taking into account some of the unique properties of BLE, mentioned in the last question?

- How does a BLE based positioning method compare on security and privacy to other positioning methods? What suggestions can be made to enhance privacy and security?

## 1.8 Acknowledgements

Thanks goes to Dr. Robert Harle for supervising this dissertation and offering guidance when I was stuck, as well as to Dr. Ramsey Faragher for explaining some of the intricacies of multi-path propagation and positioning algorithms.

# Chapter 2

# BLE radio propagation

In order to develop a Bluetooth Low Energy positioning system, it is beneficial, maybe even essential, to understand the behaviour of BLE radio signals.

When a wireless signal moves in an ideal environment, one can use the received signal strength (RSS) (or more precisely the loss of signal strength, or path-loss) to determine the distance between sender and receiver. The signal power decreases quadratically with the distance; because RSS is measured in decibel, a logarithmic scale, this means that every doubling of distance leads to a signal that is approximately 6dB weaker. When the signal strength from the sender is known (either because the sender advertises its transmit power as part of its advertising packet, or this information has been retrieved from somewhere else), the distance to the sender can be calculated. Therefore, using the RSS of multiple transmitters is a good property to use to determine position in an ideal environment.

However, in practical indoor positioning situations, there are a number of factors influencing RSS, which have to be taken into account if RSS is to be used for positioning. In this chapter I discuss several.

## 2.1 Three advertising channels, three frequencies

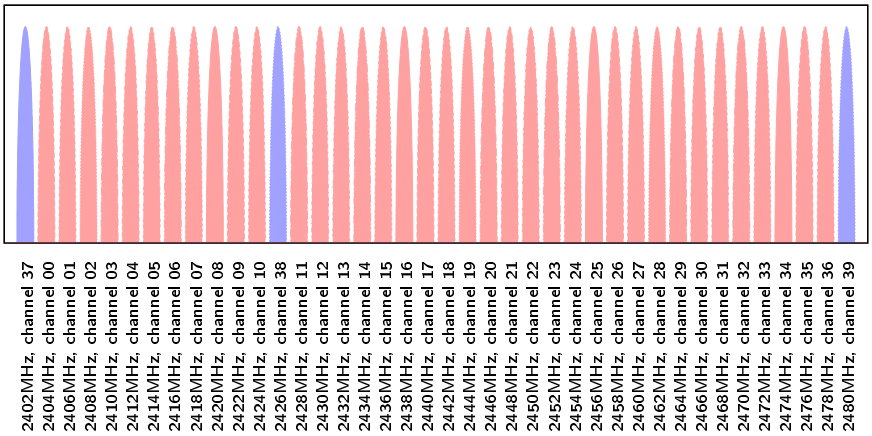
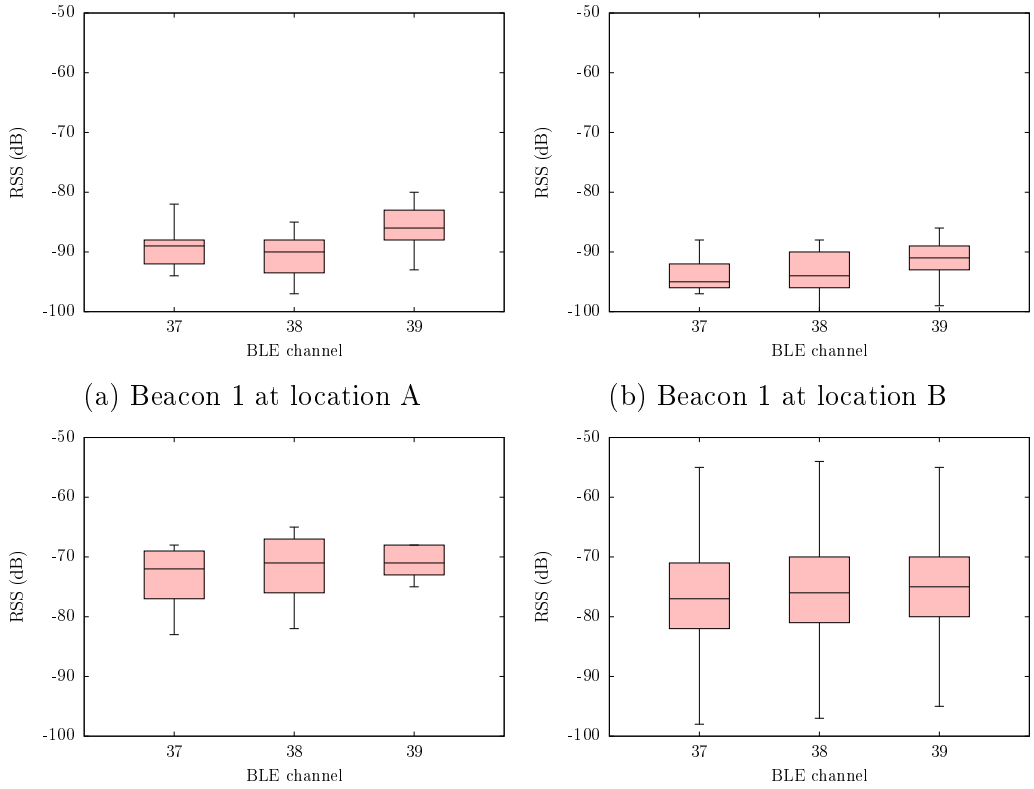



Figure 2.1: BLE frequency, with three advertising channels

Bluetooth Low Energy uses 40 channels, each 1MHz wide, in the 2.4GHz spectrum [Bluetooth SIG, 2010]. Channels 0-36 are used for BLE connections, while channels 37, 38 and 39 are used for advertising (figure 2.1). The advertising channels are used for advertisement and discovery of services, or to broadcast or unicast small pieces of information without making a connection. Once a connection is made, the advertising channels are not being used any more, instead (a subset of) the other 37 channels are used.

A possible way to build a positioning system using Bluetooth Low Energy is to use the above mentioned capability to broadcast small pieces of information on the advertising channels. The information sent is some sort of unique and recognisable beacon ID. This is the system I will focus on in this report.

As figure 2.1 shows, the advertising channels 37, 38 and 39 are distributed




(a) Beacon 1 at location A

(b) Beacon 1 at location B

(c) Beacon 2 at location A

(d) All beacons and locations combined

Figure 2.2: Distribution of RSS per channel

though the spectrum; this is done in order to maximise the chance that at least one of these channels is available in a situation with lots of radio interference. The difference in frequency between the extremes, channel 37 and 39, is 78MHz; this difference in frequency has an effect on the RSS.

Different frequencies result in different behaviours of antenna and other analogue parts in both the transmitter and the receiver, and have different radio propagation properties. Figure 2.2 shows the distribution of RSS per channel in different situations. It shows that individual beacons or locations have different RSS distribution per channel, however combining all measurements shows three almost equal distributions, small changed may be attributable to sender or receiver design, or chance.

## 2.2 Smartphone and BLE

A typical device can only send or receive on one channel at any time. The Bluetooth 4.0 specification [Bluetooth SIG, 2010] specifies that the transmitter broadcasts the same advertising packet on (a subset of) channels 37, 38 and 39, each next channel's transmission beginning less than 10ms after the previous channel. No such requirement is put on receivers, and they may use their own strategy to decide when to listen to which advertising channel. Switching between channels is preferred over listening on a single channel, since a single channel may suffer from interference and some transmitter may not be heard on that channel; something we can see in multiple occasions in section 2.5.

During the research I had four smartphones and one iPad mini in my possession; although the limited tests I did with them should in no way be considered a comprehensive research into how smartphones listen at the different channels, it does help to illustrate the differences between the phones' capabilities. Whereas no extensive test was done on the iPad mini, a quick test showed the same behaviour as for the other iOS devices.

- *iPhone 5 and iPhone 5S on iOS 7*: Both work in the same way. Listening starts at channel 37 for ∼40ms, then channel 38 for ∼40ms, then 39 for ∼40ms, before cycling back to channel 37, cycling through all three channels in 115ms. After ∼170 cycles of this, 21 seconds of listening, there is a small period of not listening between each channel switch, meaning all three channels get cycled in 180ms. After another ∼500 cycles, 111 seconds after listening starts, the period of not-listening is increased so that a full cycle through all three channels takes 890ms. I believe this is being done in an effort to save battery during long-running scans. A program can restart BLE scanning, and the process above restarts, so restarting BLE scanning every 20 seconds results in a continuously high scan rate; I did this consistently during the experiments in this report to capture as many packets as possible. iOS has extended the Bluetooth specification, to also return the channel on

which an advertising packet was received.

- *Google Nexus 4 on Android 4.4.2*: Android does not return the channel on which an advertising packet is received. Using beacons that only advertise on a single channel, it was still possible to infer information on the channel the phone was listening to at any moment. The information received shows no particular pattern in channel selection, where a packet received on channel 38 could be followed by one on channel 39, followed again by one on channel 38 all within a couple of milliseconds according to the system clock. It may be that the radio is jumping around between frequencies, or (more likely) that the timestamp reported for an advertising packet is not accurate enough to distinguish the radio jumps.
- *Samsung S4 on Android 4.3 and 4.4.2*: On Android 4.3, the phone slowly cycles between channels, a full cycle between all channels taking $\sim$ 600ms. After upgrading to Android 4.4.2, the phone exhibited the same behaviour as the Nexus 4.

## 2.3 Multi-path interference

In a typical indoor environment, wireless signals are obscured by, and reflect off, walls, objects and persons. As a result the same signal travels from the transmitter to the receiver along different paths. This phenomenon is known as multi-path propagation.

One of the ways how multi-path propagation can have an influence on RSS, is through multi-path interference. Multi-path interference occurs when a signal propagates via two paths, to be received in the same point. Depending on the length difference of the two paths, the two signals may strengthen or weaken one another (theoretically making it possible for them to cancel one another out completely).

To explore this effect I built a device that pulls a receiver (an iPhone 5) along

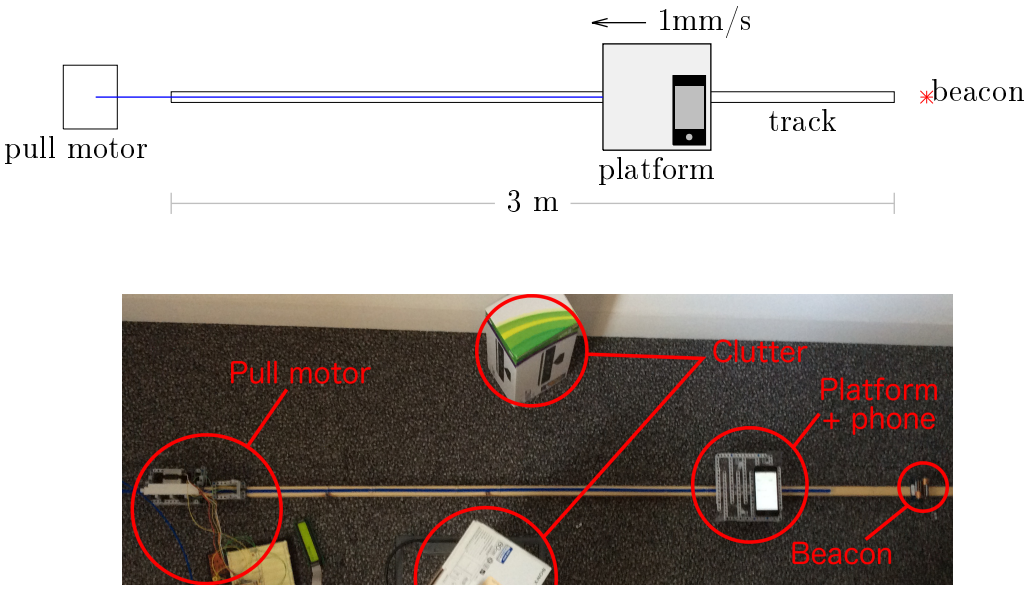



Figure 2.3: To measure the effect to multi-path interference, I built a device that pulls a platform with the smartphone slowly and precisely along a 3 meter path away from a beacon in a cluttered environment. Schematic and photo. The device in the photo is actually shorter than 3 meter, for clarity.

a 3 meter track away from a BLE beacon at precisely 1mm/s (figure 2.3). For each package received, I log the time and the RSS. Knowing the start-time and start-position, and the speed, I can plot the RSS against the position. Figure 2.4 clearly shows drops and peaks in the signal that can be attributed to multi-path interference, however at most distances hugely different RSS values are measured. Figure 2.5 shows the RSS split out by channel, and here we can see that within each channel, the RSS is fairly constant for one location, and each channel shows its own multi-path interference drops.

Figure 2.6 shows the RSS for a single channel (37) over three runs (top three graphs), and compares this to channel 38 during the third run. The same multipath drop pattern is visible in all three runs for one channel, while the other channel has a different pattern. Table 2.1 shows the Pearson correlation (using 1-second means) between different channels and between different runs. The green cells show the same channel in different runs,

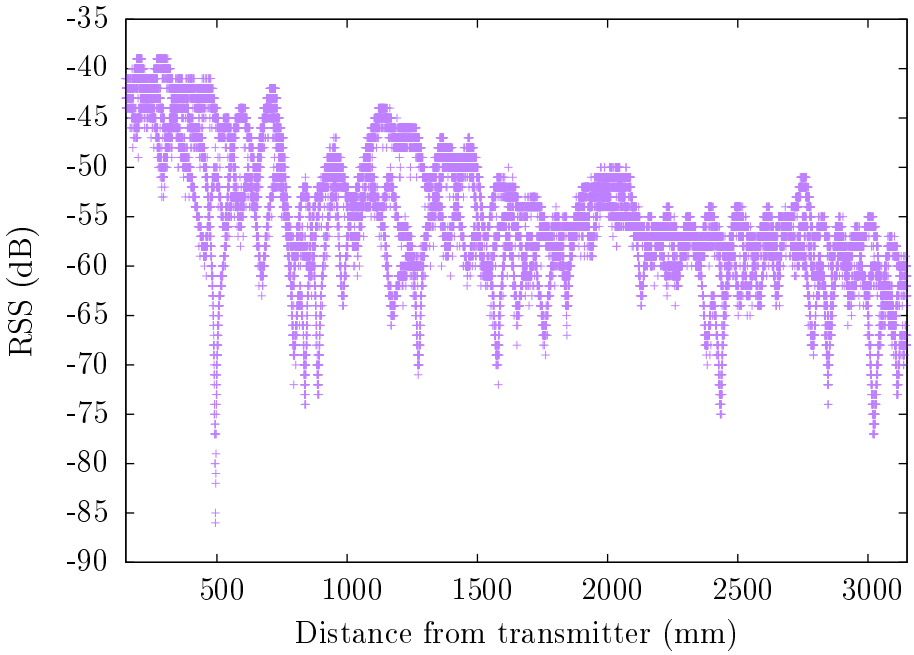



Figure 2.4: Received signal strength at different distances to a transmitter.

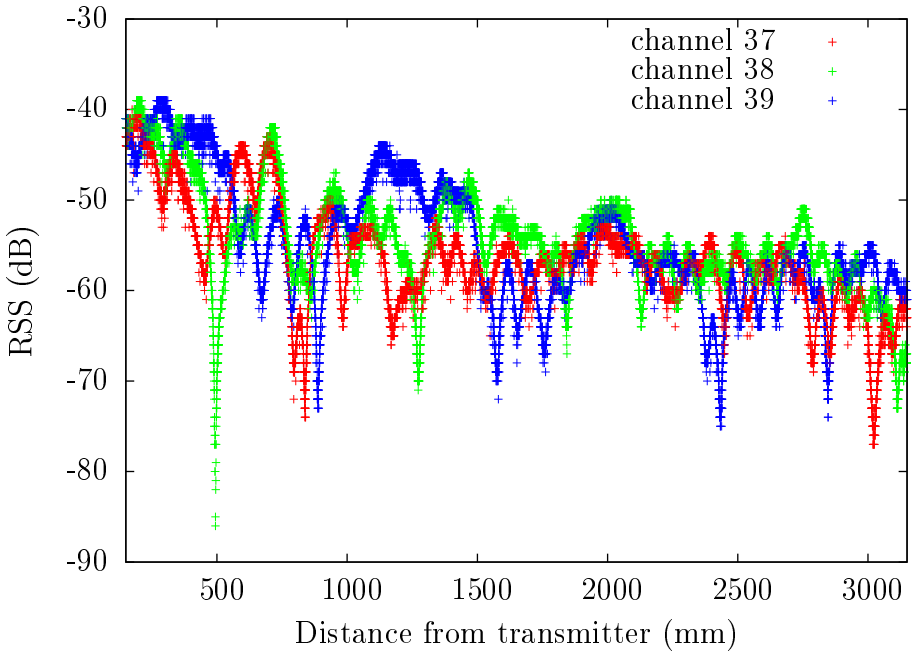



Figure 2.5: Received signal strength at different distances to a transmitter, split out per channel.

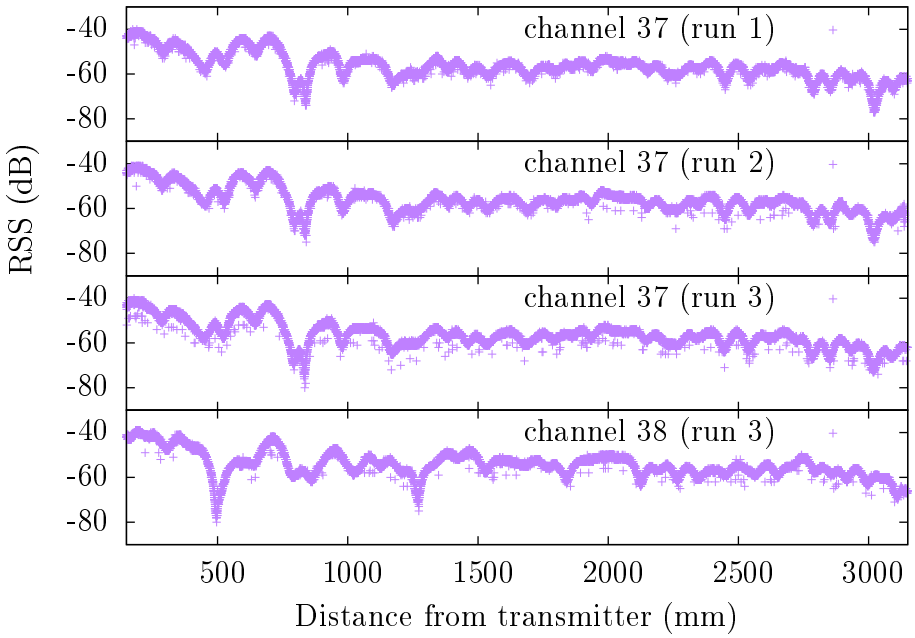



Figure 2.6: Top three graphs are RSS of channel 37 during 3 different runs. Bottom graph is RSS of channel 38 during the third run.

| | | run 1 | | | run 2 | | | run 3 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | ch. 37 | ch. 38 | ch. 39 | ch. 37 | ch. 38 | ch. 39 | ch. 37 | ch. 38 | ch. 39 |
| run 1 | ch. 37 | 1.00 | 0.67 | 0.46 | 0.97 | 0.68 | 0.44 | 0.98 | 0.63 | 0.44 |
| | ch. 38 | 0.67 | 1.00 | 0.49 | 0.67 | 0.97 | 0.48 | 0.71 | 0.95 | 0.47 |
| | ch. 39 | 0.46 | 0.49 | 1.00 | 0.43 | 0.44 | 0.98 | 0.45 | 0.48 | 0.97 |
| run 2 | ch. 37 | 0.97 | 0.67 | 0.43 | 1.00 | 0.67 | 0.42 | 0.93 | 0.62 | 0.42 |
| | ch. 38 | 0.68 | 0.97 | 0.44 | 0.67 | 1.00 | 0.44 | 0.71 | 0.91 | 0.41 |
| | ch. 39 | 0.44 | 0.48 | 0.98 | 0.42 | 0.44 | 1.00 | 0.44 | 0.47 | 0.92 |
| run 3 | ch. 37 | 0.98 | 0.71 | 0.45 | 0.93 | 0.71 | 0.44 | 1.00 | 0.66 | 0.44 |
| | ch. 38 | 0.63 | 0.95 | 0.48 | 0.62 | 0.91 | 0.47 | 0.66 | 1.00 | 0.46 |
| | ch. 39 | 0.44 | 0.47 | 0.97 | 0.42 | 0.41 | 0.92 | 0.44 | 0.46 | 1.00 |

Table 2.1: Pearson-correlation between multiple runs, on the same channel and different channels. Since many data points are involved, there is an extreme high certainty, and the p-value for all measurements is under machine epsilon.

and clearly have a much higher correlation than between different channels (the white cells). This suggests that different channels have their multi-path interference drops at different locations, something that is to be expected, since the location of the multi-path interference drops is dependent on the wave length, thus on the radio frequency.

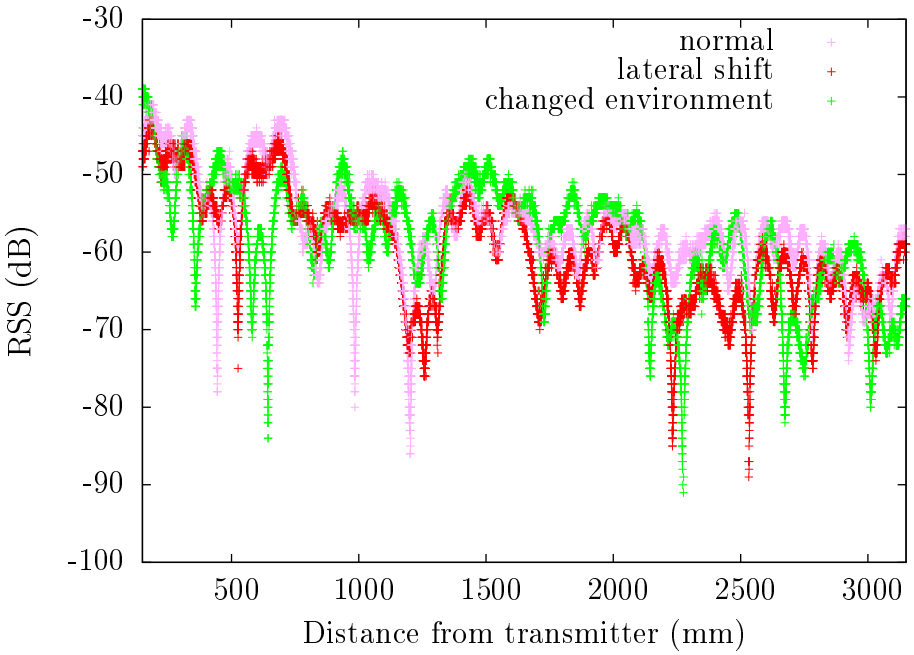
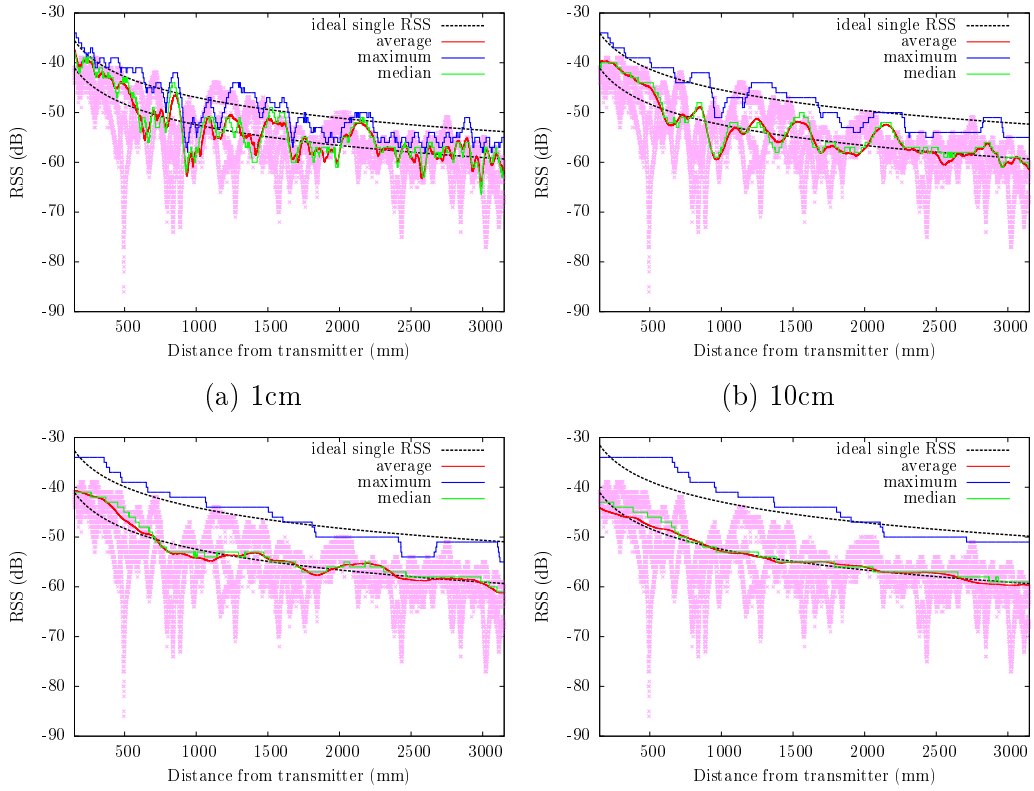



Figure 2.7: Effect of a small lateral shift of the receiver and small changes in the environment on the RSS

It is tempting to try to use the multi-path interference drops to improve on positioning via RSS; if the drops can be mapped precisely, the combination of the three individual channels could provide a location with much precision. For instance, if channel 37 has an RSS of -63dB, channel 38 has -50dB and channel 39 an RSS of -46dB, one could use the information from figure 2.5 to find a distance of 1.3m. One of the challenges here is that the multi-path interference drops are very local, and to create fingerprints for a whole room, measurements have to be taken every couple of centimetres at least in all three dimensions. To illustrate this, figure 2.7 shows in purple the original measurements, and in red the measurements taken with the receiver

put on the other side of the platform, 4cm to the side of where it was in previous runs. The same channel in both runs has a Pearson correlation of between 0.83 and 0.88, p-value $<$ machine epsilon, considerably less than the correlation between different runs with the receiver in the same spot. Even if such a map had been made however, multi-path interference is dependent on the environment, and even small changes in the environment (movement of objects or people), may invalidate a map. Figure 2.7 shows this effect, purple is the base measurement, green the measurements after some small changes in the environment are made. This time the correlation coefficient is between 0.48 and 0.66, the same as between different channels in one run. These two effects mean that simply creating a map of an environment, even for an environment with just small changes, is not feasible, unless additional techniques are developed to combat the described problems.

The fluctuating RSS value is unhelpful for most positioning methods. Most positioning methods work by creating a fingerprint map, with RSS values expected to remain more or less the same over time on the same location, and on locations between two spots where fingerprints were collected, the collected RSS values should have a value between those measured in the two surrounding mapping points. Ideally there is a smooth decrease in RSS with distance; the 6dB decrease per doubling of distance, discussed earlier in this chapter, is used, with different starting points, as an ideal reference line in this section. Figure figure 2.8 shows several strategies we may employ to remove RSS fluctuations. I take the mean, median and maximum of the received data, over different distance intervals, and compare this to an ideal RSS line. Taking this over a distance is artificial, since during positioning one does not know the distance travelled; therefore a method should be chosen that works well with any distance. Figure 2.8 shows that both average and median follow one another very closely, with sometimes one and sometimes the other being closer to the ideal line, whereas the maximum fits less nicely but is still has nice smoothing properties.




(a) 1cm

(b) 10cm

(c) 40cm

(d) 100cm

Figure 2.8: Using average, maximum and median strategies to determine RSS, using different distance intervals

## 2.4 Orientation

The direction in which a user is facing while using a smartphone to position, also has an influence on the RSS. This may be due to two reasons. First, the phone itself: rotating the iPhone leads to the antenna being less or more aligned with the signal, and leads to certain parts of the phone blocking the radio signal. Second, the user's body may either absorb or reflect the signal, leading to signal loss or gain.

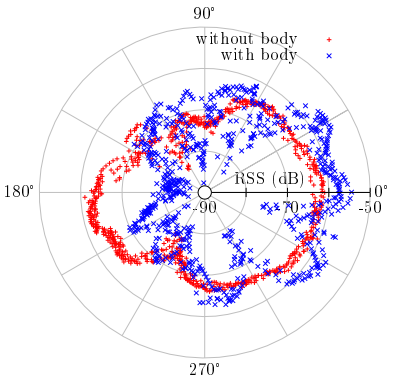
To explore these effects I built a device to rotate the smartphone around its axis: an small construction was added to the back of an office chair, such that a smartphone rotates around its axis when the chair rotates (figure 2.9).

(a) without body (b) with body

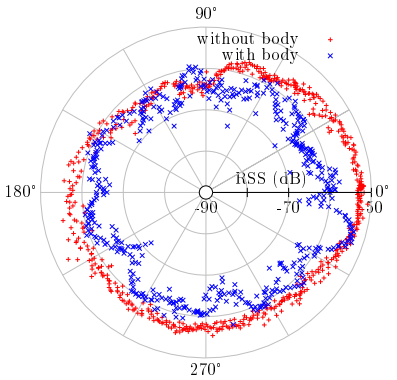
Figure 2.9: Device used for measuring rotation.

Using the built-in compass, the heading was recorded together with the RSS. During measurements an eye was kept on the reading from the compass and no large errors were spotted; small errors in compass accuracy do not have a major influence on the results of this section. Two measurements were made at each location. In 300 seconds I rotated the chair 3 $\times$ 360°, while measuring compass heading, and the RSS for a beacon placed at around 7 meters distance, while lying on the floor under the chair. Then I sat on the chair with my body around 15 cm from the smartphone, and did the same 1080° rotation in 300 seconds.

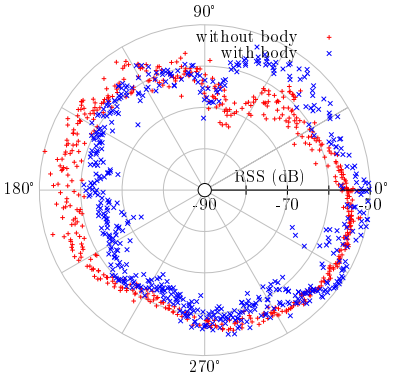
Figure 2.10 shows in red the average RSS per 0.5 second, plotted against the direction the phone is facing, while I lie under the chair. 0° means that the back of phone is the facing the beacon, while at 180° the screen is facing the beacon. Since on the iPhone 5 the Wi-Fi/Bluetooth antenna



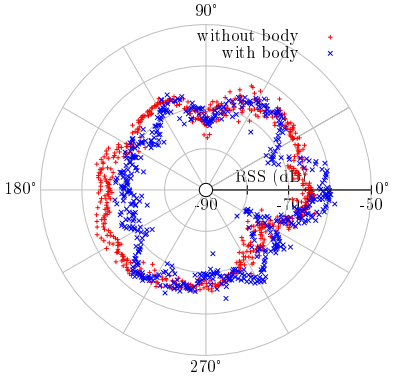

(a) Outside in a field




(b) Indoor in a mostly open room




(c) Indoor in a corridor




(d) Indoor, transmitter and receiver in different rooms

Figure 2.10: RSS under rotation, with and without a body present, in different environments.

is mounted against the back of the phone, next to the camera and flash[1], I expect reception to be best with that side facing the transmitter. The blue points show the same rotation with my body on the chair, influencing the signal. Since I'm facing the screen, at 0° the smartphone is between me and the beacon, while at 180° I am between the beacon and the smartphone.

Figure 2.10(a) shows that without a body present, both with the back and with the screen facing the beacon, there is optimal reception of -63dB. Reception generally stays above -70dB, however at two points, around 120° and 240° there are large drops to around or even under -80dB, resulting in an

[1] `http://www.ifixit.com/Guide/iPhone+5+Wi-Fi+Antenna+Replacement/10897`, accessed on 11 June 2014.

RSS difference of 15dB in just 15°, and a 20dB difference throughout the whole rotation. These plotted values are already averages, the extremes on different channels are even larger.

In the blue points, the effect of the body is clearly visible, shadowing the signal, resulting in a 20dB drop when between the phone and the beacon, while resulting in a slight boost of the signal when the back of the phone faces the beacon, possibly because of the reflection of the signal by the body. Also in this case we see 20dB+ drops in the signal strength within a couple of degrees in many spots.

In figure 2.10 the experiment is repeated in different environments, generally coming less open from (a) to (d) In general in (b), (c) and (d) the same patterns as in (a) are visible, that at 180° the body casts a shadow on the reception, while at 0° a slight boost of signal can be seen when there is a body behind the device. The effect is arguably smaller in the more cluttered environments, possibly because reflections result in multiple paths the signal can take around the body. All environments show some drop orientations, where the RSS drops by 10-20dB within a couple of degrees.

## 2.5 People moving through the room

In order to investigate how people and objects moving around a room influence RSS, 6 beacons were placed in a meeting room, together with an iPad mini as receiver on a table (figure 2.11), while a standing lunch was taking place. All tables and chairs were moved to the side before the start of measurements. Figure 2.12 shows the RSS over time for six beacons, and how many packets were received on each channel. Even though beacon 6 was furthest away, the highest RSS of all beacons is received from this one; this may be because it is in front of the receiver on a direct line of sight.

Five periods were marked using a wristwatch, these are shown in the graphs:

- Period 1 (12:41-12:59): The lunch is being built up, occasionally people

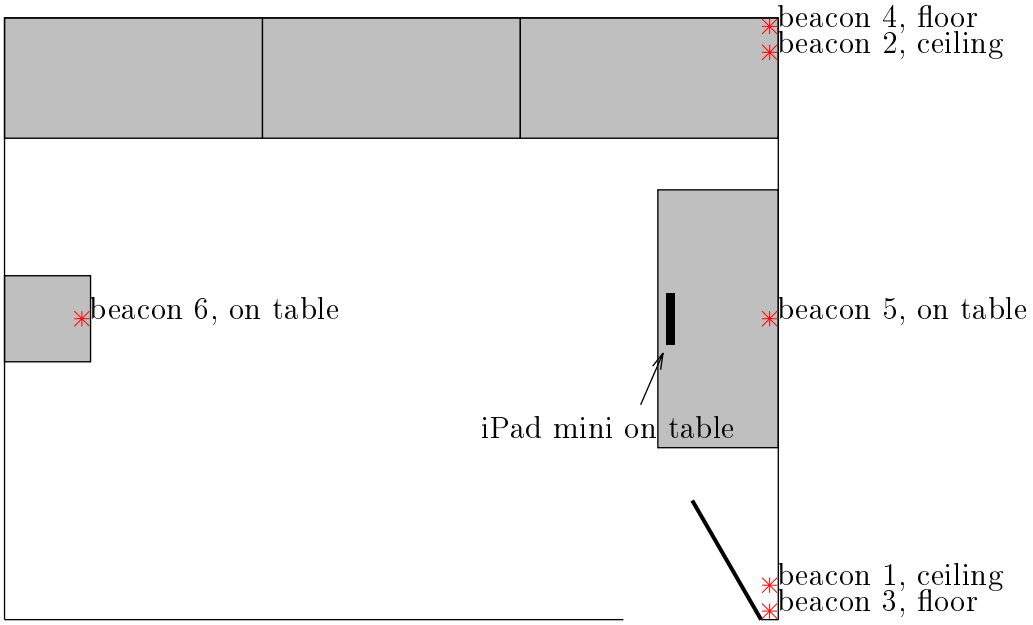
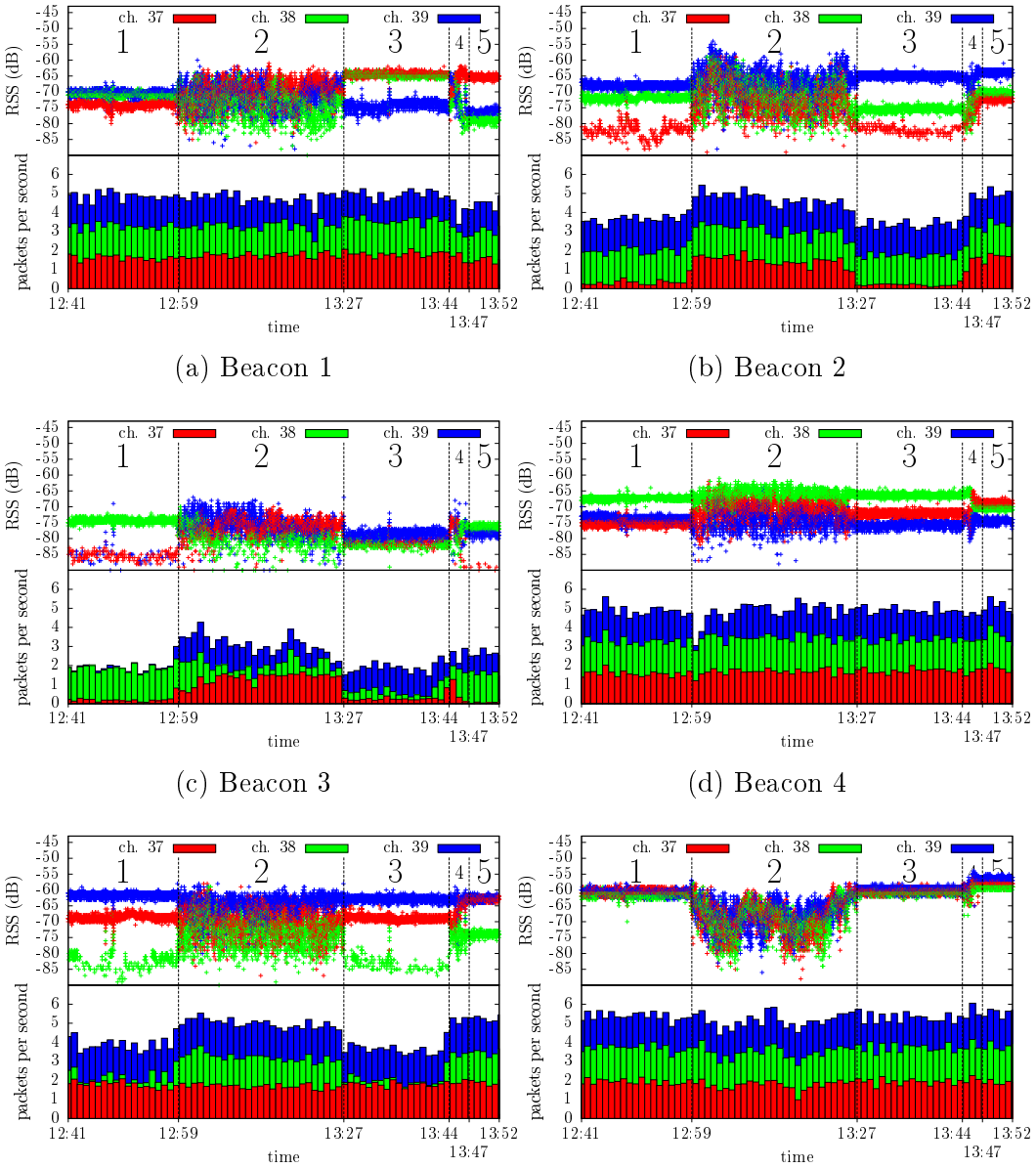



Figure 2.11: Schematic of meeting room in lunch setting.

enter and leave the room, most of the time there are one or two people in the room. The door to the room is open.

- Period 2 (12:59-13:27): The room fills up with people, people move around, in the first half of the period there are between 15 and 20 people in the room, after which the room slowly empties. At the end of the period the last group of 3 people leave, and they close the door.
- Period 3 (13:27-13:44): The room is empty.
- Period 4 (13:44-13:47): Without moving any beacons or receivers, the furniture in the meeting room is put back to meeting room position, the door is closed again afterwards.
- Period 5 (13:47-13:52): The room is empty.

The two periods with an empty room, 3 and 5, are characterised in the graphs in figure 2.12 by relatively low variance in RSS per channel, whereas period 2, when a lot of people are moving around, shows high variance,




(a) Beacon 1

(b) Beacon 2

(c) Beacon 3

(d) Beacon 4

(e) Beacon 5

(f) Beacon 6

Figure 2.12: RSS and the number of packets received for the six beacons during different room occupation.

with periods 1 and 4 being somewhere in between. The differences between period 3 and 5 likely show the influence of the furniture on the RSS, where major differences between period 1 and 3 are most likely caused by the door being open or closed. This latter effect is especially clear with beacons 1 and 3, which are respectively in the ceiling and on the floor behind the door. Another interesting effect is that only in beacon 6 all channels become lower during period 2. Possibly this is because for beacon 6 the line-of-sight got interrupted during period 2, while most other beacons did not have a line-of-sight, and sometimes benefit from the extra reflection possibilities of a room full of people.

The beacons broadcast at 9.5Hz, however only once more than 6 advertising packets per second are received on average during a minute. In the next section I look into packet loss in more detail, but it is interesting to note the large differences in packet loss between channels, and between periods. It shows that when the RSS drops below -80dB, fewer (or sometimes hardly any) packets are received.

## 2.6 Packet loss

According to the Bluetooth 4.0 specification, an advertising packet should be sent on all three advertising channels within a short time. This means that a beacon advertising at 10Hz (the beaconing rate) sends 10 packets per second on each of the three channels, heaving a 100ms *beaconing interval*.

A smartphone typically listens on one channel at a time for BLE packet, switching between the three channels. We can therefore assume that, even though a 10Hz beacon sends 3 (channels) $\times$ 10 packets per second, a smartphone will at most receive 10 of these per second; any fewer would be considered packet loss. There are many reasons why a smartphone may receive fewer than 10 packets, for instance: the signal strength is too low to be received, other transmitters interfered with the signal, the smartphone was temporarily not ready to receive packets, or the OS discarded a packet.

As mentioned in section 2.2, an iOS device misses more packets if Bluetooth Low Energy listening is active for a longer time. To counter this, I restart BLE listening every 10 seconds in all experiments; I do not see increased packet loss at the time of restarting.

If packets are not received consistently, this may impact the positioning ability. There is no way for the device to know whether a beacon was not observed because it is out of range, or because the packets sent by the beacon were lost. To counter this, a smartphone has to scan for a number of beacon intervals; especially if beacons advertise at a low rate to save energy, this may be a problem.

To investigate packet loss I used a single beacon transmitting at 1Hz. The measurements were done outside in a field, with no other known transmitters nearby, distance between beacon and smartphone 15 centimetres. In 136 beacon intervals, 104 packets were received, a 23.5% packet loss.

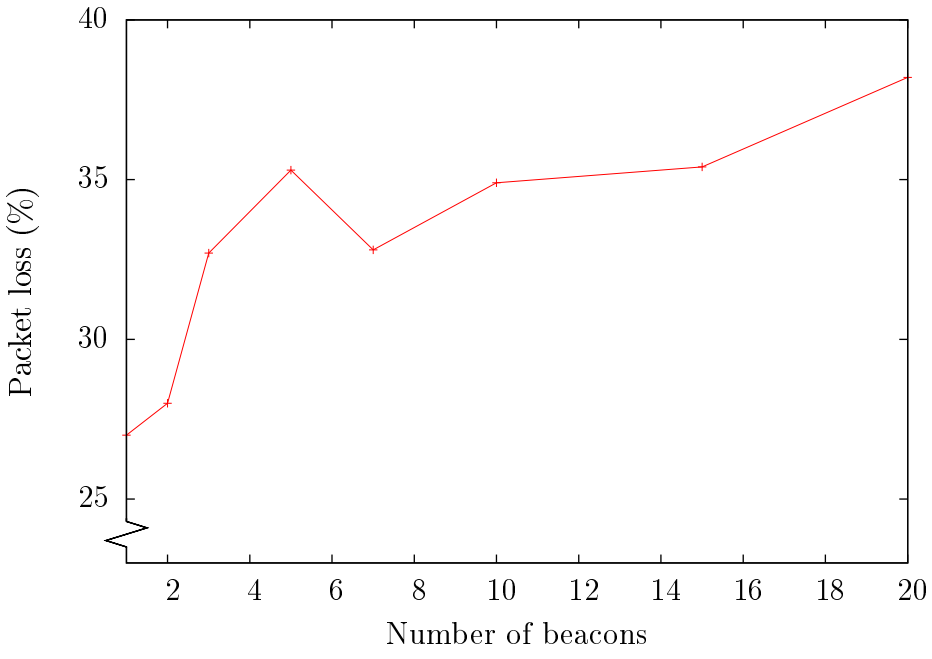



Figure 2.13: Packet loss for different numbers of beacons.

For positioning it is important to know how long one has to listen to be rea-

| | listening time in transmission intervals | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 1 beacon | 27% | 4.6% | 0.55% | 0.18% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| 2 beacons | 51% | 11% | 1.4% | 0.14% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| 5 beacons | 88% | 48% | 21% | 8.4% | 3.1% | 0.73% | 0.00% | 0.00% | 0.00% | 0.00% |
| 10 beacons | 99% | 74% | 35% | 15% | 6.0% | 2.6% | 0.92% | 0.37% | 0.18% | 0.00% |
| 15 beacons | 100% | 85% | 52% | 26% | 13% | 6.1% | 2.6% | 0.92% | 0.18% | 0.00% |
| 20 beacons | 100% | 96% | 73% | 45% | 24% | 11% | 5.7% | 3.2% | 1.9% | 1.2% |

Table 2.2: Measured chance that not all beacons have been observed at different listening intervals.

sonably sure that, if a beacon has not been observed, it can not be observed in that location. I repeated the experiment with first a single beacon set to transmit at 10Hz[2], then multiple beacons, all set to 10Hz, all within a 20cm radius of the smartphone. Each test was ran for one minute; the results can be seen in figure 2.13. It shows a packet loss of between 27 and 38%, with a generally a higher packet loss for a larger number of beacons. This can be explained because packets from multiple beacons may collide when sent at the same moment.

Table 2.2 shows the results from the same experiment, but considers how often multiple subsequent packets from the same beacon were lost. It shows that with one beacon, in 0.55% of the cases I had still not received it after three beacon intervals. With twenty beacons, after 10 beacon intervals, in 1.2% of the cases I had not observed each beacon as least once.

The tests above were done in an area with minimal additional radio interference. Looking at a more noisy environment, such as the one of section 2.5, there is 59% packet loss, and even after 5.25 seconds, 50 beacon-interval at 9.5Hz, in 1% of cases not all beacons have been observed. This can probably be attributed to interference from Wi-Fi and other Bluetooth devices, as well as there not always being a line-of-sight between beacon and smartphone.

[2]The actual beacon rate of these beacons is 9.5Hz. In the text I will continue to call them 10Hz beacons, because 10Hz and 100ms intervals are easier to reason with, and because 10Hz is the setting the beacons were set to. All packet loss calculations in this section have been done with the actual rate of 9.5Hz.

# Chapter 3

# BLE positioning in practice

## 3.1 RSS based positioning

Bahl and Padmanabhan [2000] laid the foundation for positioning using Wi-Fi signals. The basis is a fingerprint database, containing fingerprints (the RSS, or RSS distribution, for each access point) for a large number of locations. This database can either be filled empirically, surveying the RSS at each location; or calculated, where the RSS at each location is estimated using the positions of the access points and a radio propagation model for the environment. They show that empirically filling the database leads to better results, however it may take much longer since a manual survey of each location is needed.

To do positioning, a measurement of the RSS of each access point at an unknown location is made, and each fingerprint in the database is compared to this measurement in order to find one or multiple close matches. The method mostly used to find a match is (a variant of) k-nearest-neighbour. The method works in three steps.

Firstly the set of measured RSS are transformed into values needed for step two. Bahl and Padmanabhan [2000] described that the mean, standard deviation and median of multiple measurements at a single location were cal-

culated, but only used the mean in the rest of the paper.

Secondly the values from the fingerprints are compared to the values found while positioning, and a distance between the two sets is determined, by a distance-function $L$. The usual choice for the distance-function (and hence the name) is to calculate the distance in signal-space, where each signal-source is a dimension. Li et al. [2005] described a generalised distance function (equation 3.1).

$$L_q = \left( \sum_{i=1}^{n} |p_i - s_i|^q \right)^{\frac{1}{q}} \tag{3.1}$$

Here $n$ is the number of signal-sources, $p_i$ the measured RSS for a source during positioning, and $s_i$ the surveyed RSS at the point to which the distance is to be calculated; different $q$ lead to different distance functions with $L_1$ being the Manhattan distance, and $L_2$ the Euclidean distance. There seems to be no clear consensus on which $q$ gives the best result, with Shin et al. [2012] using $L_1$, Bahl and Padmanabhan [2000] using $L_2$, only saying that alternatives (possibly also of another form) were briefly experimented with, and Li et al. [2005] using $L_1$, while noting that the difference with other $q$ values is not significant. Most methods use the $L_q$ function with the $s_i/p_i$ in dB, Li et al. [2005] explores whether these values should alternatively use the power $P$, $1/P$, $1/P^2$ or $1/P^4$, concluding that dB works the best, but $1/P^2$ and $1/P^4$ also give good results. Another interesting candidate is the Mahalanobis distance, use of which is explored and found to be superior by [Kaemarungsi, 2005].

Finally the calculated distances are being used to map to a position. Bahl and Padmanabhan [2000] uses both a 1-nearest-neighbour and a k-nearest-neighbour approach, showing that the second works better. Li et al. [2005] uses a weighted-k-nearest-neighbour approach, where the nearest neighbours are being weighted by the result of the distance-calculation, and Shin et al. [2012] introduced using a dynamic value for the number of neighbours $k$ to further improve the result.

Pandya et al. [2003] compares nearest-neighbour to other positioning meth-

ods based on RSS, and finds nearest-neighbour to work well in most cases (among which Bluetooth Classic). Unsurprisingly, methods from the domain of machine learning have been examined as well to do positioning, such as Gaussian Process Latent Variable Models [?, ferris2007wifi]r Neural Networks [?, battiti2002location]

Of these methods, the one used by Bahl and Padmanabhan [2000] is used in this report to compare against; both because the method is simple to understand and implement while giving good results, and because many other authors used this method as a base to compare against.

## 3.2 Experiment

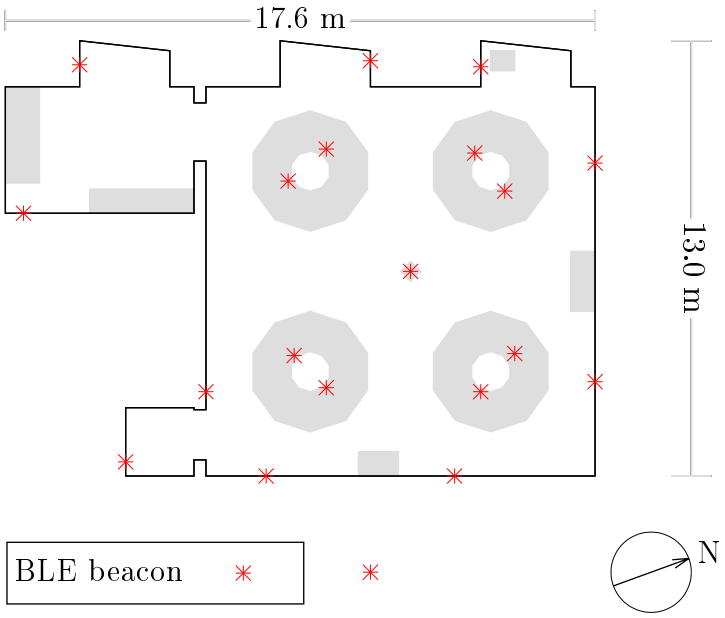



Figure 3.1: Test bed: room SW02 in the Computer Laboratory of the University of Cambridge.

Room SW02 in the Computer Laboratory of the University of Cambridge is the test bed for this experiment. The room consists of a square 12 by

12 meter main area, with several coves. The room contains four decagonal-shaped tables with computers on them and chairs around them, as well as some other furniture. Twenty BLE beacons were used, ten on the walls of the room, eight on the tables, one in the middle and one placed a couple of meters outside the room (figure 3.1). The beacon positions were chosen such that all areas would receive BLE signals, but no specific action was taken to find optimal positions. Each beacon broadcasts a single advertising packet, a unique beacon ID, on all three channels, with a rate of 10Hz. All measurements were done on a 60 by 60 cm grid, since this is the size of the floor tiles and this allowed for easy reference.

Figure 3.2: Device used for surveying.

During the surveying phase, each accessible point on the grid is surveyed, by slowly moving the smartphone along a circle with a radius of 15 cm centred on the grid point, the back of the phone facing outwards, with a human body on the opposite side of the circle, facing the screen; the normal position for a user using the smartphone. To accomplish this consistently, I adjusted the office-chair set-up of section 2.4, such that the smartphone now rotates along a circle instead of its axis (figure 3.2). For each advertisement packet received, the beacon ID, advertising channel, RSS and the current heading as

reported by the smartphone's compass, is saved. Complete 360° survey took between 3.5 and 15 seconds for each of the 226 accessible points, with a mean of 9 seconds per point. On average 903 advertising packets were captured per point, meaning there was around 45% packet loss.

During the positioning phase, I went to each grid point, standing naturally while holding the smartphone still over the grid point for two seconds. In total 670 positionings measurements were taken at 226 locations.

During surveying each beacon was observed at least once at each location. During positioning, this was not the case; this may have been because the smartphone was not moving, not rotating, and the listening interval was shorter.

In addition the $x$, $y$ and $z$ coordinates for each beacon were determined, and the maximum RSS on 1 meter distance.

The data gathered from this experiment allows me to compare six different positioning algorithms, and simulate how they perform under different circumstances.

## 3.3 Positioning methods

All positioning in this report is being done through the weighted k-nearest-neighbour algorithm. Provided a set of distances for a single measurement to each surveyed point, ordered by distance, the average of the points belonging to the first $k$ distances is calculated, weighted by $\frac{1}{distance+\epsilon}$. The difference between the compared algorithms, is in how the distances are calculated, the first two steps of the nearest-neighbour algorithm as described in section 3.1. All experiments are run with $k = 5$; this value was empirically found to work well.

In the positioning algorithms below, the first step is always (except for "random") taking either the mean or the maximum RSS measured at a location. To deal with cases when a particular beacon was not observed, that beacon

is treated as having been received with an RSS of -105dB; this is 2dB lower than the lowest RSS reported for an actual packet during the experiment, and seems reasonable, since if there had been a packet with signal strength -105dB, the smartphone would not have noticed it.

### 3.3.1 Random

As a reference, the random method returns random distances for each fingerprint, after which the k-nearest-neighbours method is used to calculate a position.

### 3.3.2 Signal Space Distance (SSD)

The method described in Bahl and Padmanabhan [2000], using dB for the RSS, and the $L_2$ distance function (equation 3.1). As in Bahl and Padmanabhan [2000], the mean RSS over all received packets is used both in the surveying and the positioning data. Only one fingerprint per position is taken, unlike Bahl and Padmanabhan [2000], who created four fingerprints in four different directions.

### 3.3.3 Signal Space Distance with Orientation (SSD-O)

In section 2.4 I showed that the direction of measurement has a huge influence on the RSS. Bahl and Padmanabhan [2000] recognised this in Wi-Fi, and surveyed in four directions; during positioning however they had no way to determine the orientation, and all four directions were considered equally during positioning. King et al. [2006] considered how knowledge of the orientation improves positioning. They surveyed in eight directions, and considered different ways of combining these into ad-hoc fingerprints during positioning.

I consider a variant of King's method, using the Signal Space Distance, but

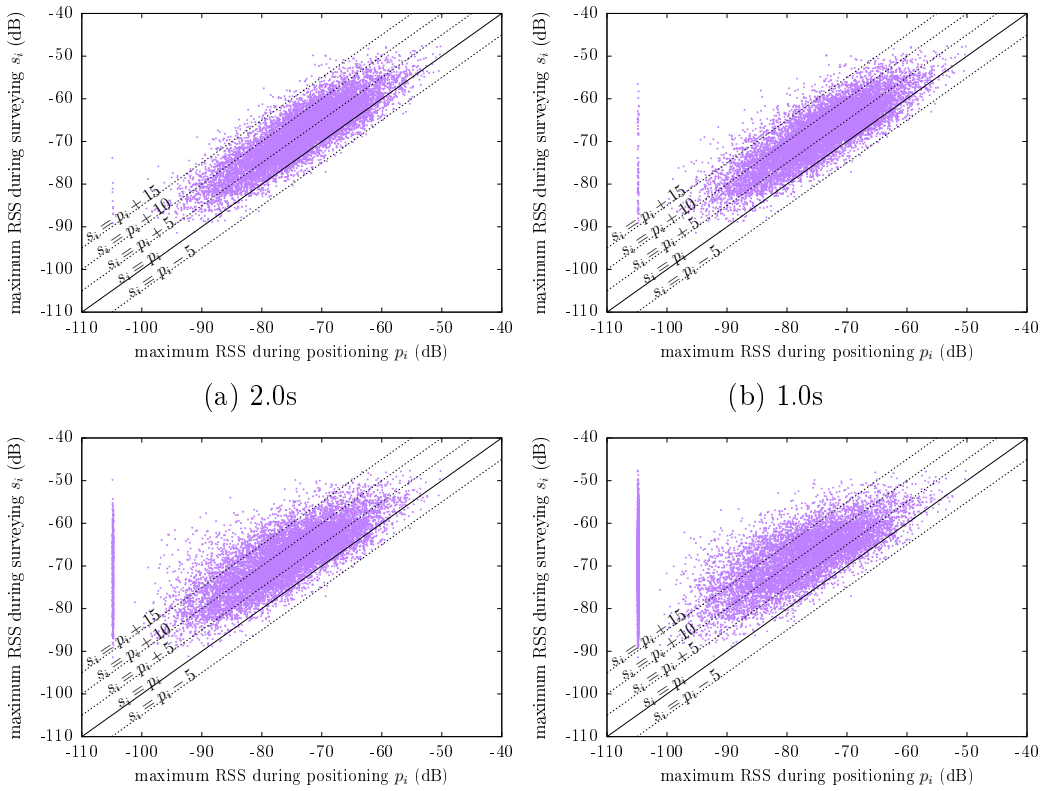
taking orientation into consideration: SSD-O. Unlike the Wi-Fi surveying used by Bahl and Padmanabhan [2000] and King et al. [2006], I survey continuously while turning 360°. For each packet received, the heading as reported by the smartphone's compass is stored in the database, with the beacon ID and the RSS. While positioning, the smartphone is expected to not be rotating much; the average heading $h_p$ of the device is used to build an ad-hoc fingerprinting database. This database is built using the mean RSS, where at each location, for each beacon, only those packets are considered that were received during surveying with a heading $h_s$ where $h_p - \alpha < h_s < h_p + \alpha$. A value of 60° for $\alpha$ was found to work best (see section 3.4). In order to not have to calculate the ad-hoc fingerprint database for each positioning, the experiment was run with 360 pre-calculated fingerprint databases, one per 1°, where for each positioning the closest database is used. For example, if the average heading during positioning is 34.6°, the pre-calculated database at 35° will be used, which is built using all surveying packets that were received between 335° and 95°.

It should be noted that this method is dependent on the accuracy of the built-in compass. During surveying and positioning I kept an eye on the compass reading, and noticed nothing absurd, however no further effort was made to check how accurate the compass is. In section 3.4.3, I explore the influence that a potential compass-error has on this method.

### 3.3.4 Blackout Resistant Positioning (BRP)

I introduce Blackout Resistant Positioning to do positioning with Bluetooth Low Energy, taking into account some of the radio propagation properties of BLE that were described in chapter 2. Most notably, the method recognises that some beacons may be temporarily blacked-out; they may be received at much lower RSS or not at all due to multi-path interference, orientation, body shadowing or packet loss.

BRP uses the same k-nearest-neighbour approach as the previous methods, however since it does not calculate distance between a fingerprint and the




(a) 2.0s (b) 1.0s

(c) 0.5s (d) 0.2s

Figure 3.3: Relation between the maximum RSS per beacon measured in a spot during surveying and positioning, for different listening times. If a beacon is not observed, it is assigned an RSS of -105dB.

measurement in a multi-dimensional signal space, it intuitively makes more sense to talk about a *penalty function* in this context. The function gives penalty points for each way in which a fingerprint does not agree with a positioning measurement. This syntactic difference does not change its semantics however: penalties are calculated for each fingerprint in the database, and the ones with the lowest penalties are selected to be averaged.

Blackout Resistant Positioning recognises that, once during surveying the maximum RSS $s_i$ in an area has been determined for a certain beacon, it is unlikely that during positioning a much higher RSS $p_i$ will be measured at that point. This means that $p_i \gg s_i$ should result in a large penalty. On the other hand, having measured a much lower RSS may be the result of multi-path interference, antenna orientation, body shadowing, or packet loss, and is not a strong indicator that a particular position should be discarded; hence $p_i \ll s_i$ should not carry a large penalty. Obviously the lowest penalty is given if $p_i \approx s_i$. By adding the individual penalties given per beacon together, the full penalty for a fingerprint is obtained.

Figure 3.3 shows how $s_i$ relates to the $p_i$ on the same location. The figures show that $p_i$ is on average a couple of dB lower than $s_i$; it shows that my assumption that $p_i$ is not much higher than $s_i$ is correct, and shows that there are many situations where $p_i \ll s_i$. Shorter listening intervals lead to a higher difference between $s_i$ and $p_i$, which is to be expected since $p_i$ is the maximum for all packets, and a shorter listening period will give a smaller maximum on average. The points on the left side, where $p_i$ = -105dB indicate those situations when a certain beacon was not observed during the listening interval.

The following penalty function was empirically found to work well. It is based on $d_i$, which is the difference between $p_i$ and $s_i$. A positive $d_i$ indicates that the measured RSS during positioning is larger than the RSS surveyed

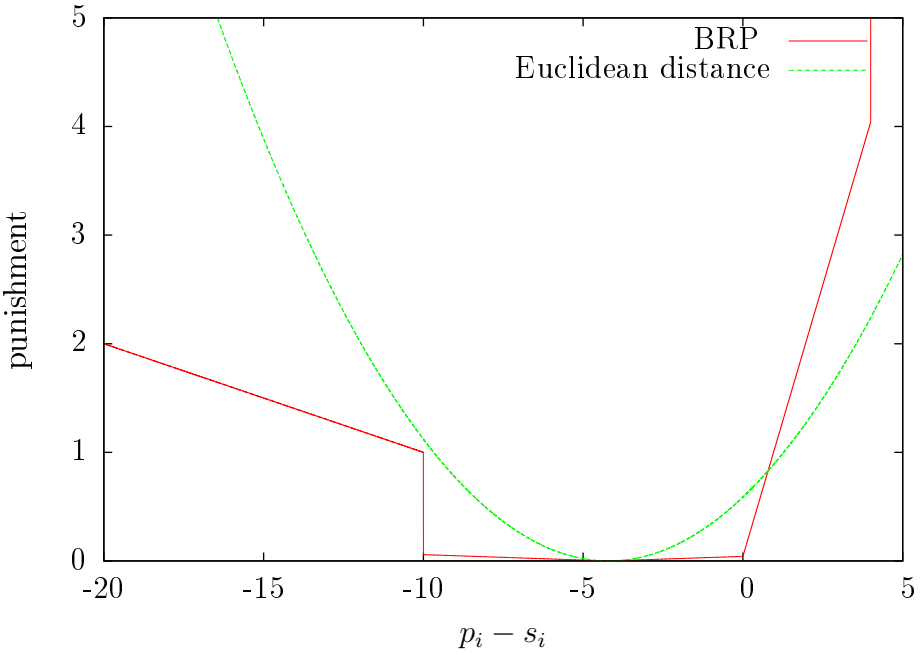



Figure 3.4: Penalty function for BRP, as compared to SSD

for a candidate location.

$$L = \begin{cases} |d_i| \times 100000 + 3.042 & \text{if } d_i > 3dB \\ |d_i| + 0.042 & \text{if } 0dB < d_i \leq 3dB \\ |d_i + 4.2| \times 0.01 & \text{if } -10dB < d_i \leq 0dB \\ |d_i| \times 0.125 & \text{if } d_i \leq -10dB \end{cases} \tag{3.2}$$

This penalty function is shown in figure 3.4 and compared to the (scaled) Euclidean distance function.

The function can be understood by realising that for $d_i > 0dB$, large and very large penalties are given (top two lines). For $d_i < -10$dB, penalties start with a large offset, but increase only slowly as $d_i$ decreases. The part in between has very low penalties with a minimum of 0 at $d_i = -4.2$. This is the average $d_i$ for the 2 second listening interval.

The function has been optimised for performance on the 2 second listening

interval, however while keeping performance in other situations in mind as well. Considering the changes in $d_i$ distribution for different listening intervals, as shown in figure 3.3, optimising for other listening intervals will probably lead to other parameters.

### 3.3.5 Blackout Resistant Positioning with Orientation (BRP-O)

Taking orientation into account in BRP, in the same way that SSD-O does for SSD, is an obvious extension. It should be noted however that BRP is specifically designed to have a certain resistance to signal drops, including those from orientation.

In SSD-O, $\alpha = 60°$ was found to work best; in BRP-O $\alpha = 90°$ works best.

### 3.3.6 Blackout Resistant Positioning with Radio Propagation Model (BRP-RPM)

Whereas BRP uses a surveyed database of fingerprints, BRP-RPM estimates the maximum RSS on a specific location using a radio propagation model. Bahl and Padmanabhan [2000] describe a radio propagation model used for Wi-Fi signals, which uses the layout of the building, and a Wall Attenuation Factor to estimate RSS at different locations. Such a model does not work well for BRP however, since BRP expects the fingerprints to be the maximum receivable RSS in that area, and the model in Bahl and Padmanabhan [2000] tries to have a low average error. Instead I use a simplified model for maximum expected RSS at a position.

$$R(x) = R_0 - 10 \times {}^{10}\log(x^2) = R_0 - 20 \times {}^{10}\log(x) \tag{3.3}$$

By measuring the RSS at one meter for each beacon (if all beacons are similar, this only has to be measured for one beacon), this formula can be used to

estimate the maximum RSS at any location.

## 3.4 Measurements

Figure 3.5 shows the RSS for a single beacon in the different fingerprint databases. The RSS looks as expected; RSS generally decreases with distance to the beacon. The SSD-O fingerprint has slightly higher averages than the SSD south of the beacon, and slightly lower north of the beacon, because of body shadowing. The maximum RSS (as shown in (c)) is about 10dB higher than the mean, with BRP-O showing the same slight directional properties.

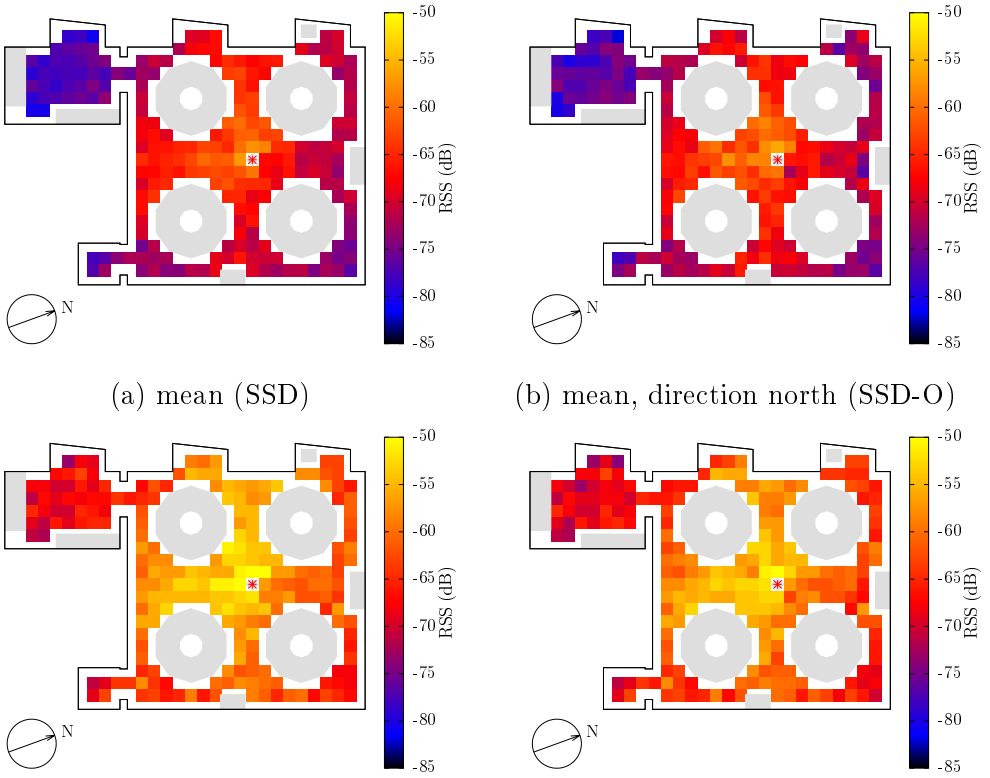
### 3.4.1 Random

Using a (deterministic) random distance function gives a mean error of around 6 meters, something that was expected considering that the k-nearest-neighbour algorithm will have a tendency to choose positions in the middle of the room as it averages $k$ positions. Obviously this would grow with the size of the test bed; it does give a lower bound for positioning.

### 3.4.2 Signal Space Distance

SSD gives a median error of 1.08 meter, and counts as the positioning function the other ones are evaluated against. Figure 3.7 shows that some areas in the room have a larger mean error than other areas, while the arrows show no obvious direction in which the errors lie.

In general positioning seems to be more accurate close to the walls; one explanation may be that orientation and multi-path interference influence the reception less close to the wall than in the middle of the room, because reflection off the wall compensates for that. I have been unable to find evidence for this however.




(a) mean (SSD)

(b) mean, direction north (SSD-O)

(c) maximum (BRP)

(d) maximum, direction north (BRP-O)

Figure 3.5: Each method uses its own fingerprint database; shown the RSS of the central beacon in each database

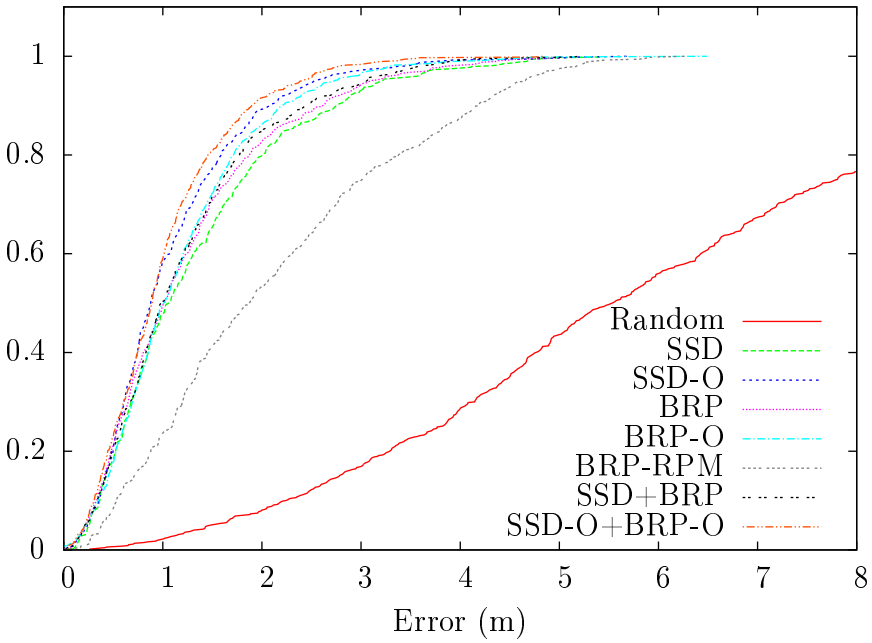



Figure 3.6: Errors for the six positioning methods and two combined methods

### 3.4.3 Signal Space Distance with Orientation

Figure 3.6 shows that using the internal compass to take orientation into account can improve positioning performance; in the experiment the error was between 10%-20% lower for SSD-O versus SSD, and even more for the upper percentiles.

The average heading stored for all packets received at a single positioning is calculated (since the smartphone is held still during positioning, all spread in heading is due to noise), and a pre-calculated fingerprint database is used which only considered all surveyed packages in an angle of $\alpha = 60°$ to the left and right of this heading. In total, 360 SSD-O databases were pre-calculated, one per 1°.

Figure 3.8 shows how the median error changes, if $\alpha$ and the number of databases are adjusted. It shows that an $\alpha$ of 45° or 60° performs best, however all databases between 30° and 90° perform close to one another. Intuitively one may expect databases with smaller $\alpha$ to perform better, how-

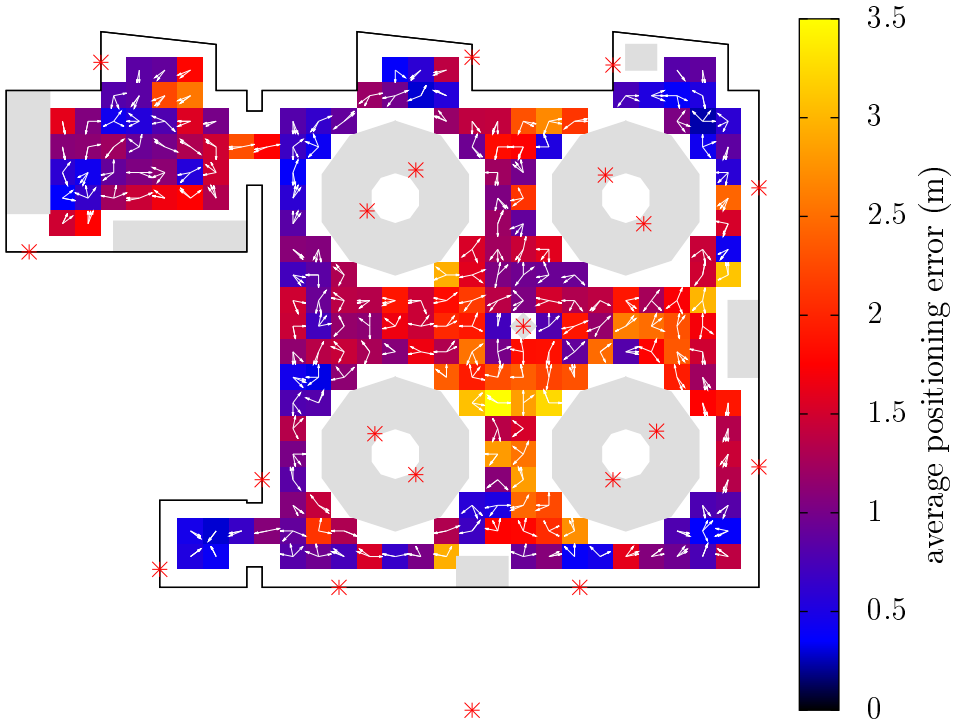



Figure 3.7: Positioning errors using SSD, colour indicates average error, arrows indicate the direction where the algorithm calculated the smartphone to be.

ever each location was only surveyed for 9 seconds on average, meaning that each 30° part was only surveyed during 0.75 seconds, or 7.5 beacon intervals, on average. Section 2.6 showed that in 7.5 beacon intervals, there is still a reasonable chance that a certain beacon was not observed; and even if all beacons were observed, multiple packets are preferred to come to a good mean. Indeed 1% of values in the SSD-O fingerprint database for $\alpha = 15$ is -105dB.

A single fingerprint database is in the order of 20 bytes per $m^2$ (see section 3.7); if we need 360 databases (one per °), this increases to 7 kilobyte. Figure 3.8 shows that more databases generally improve performance, however espe-

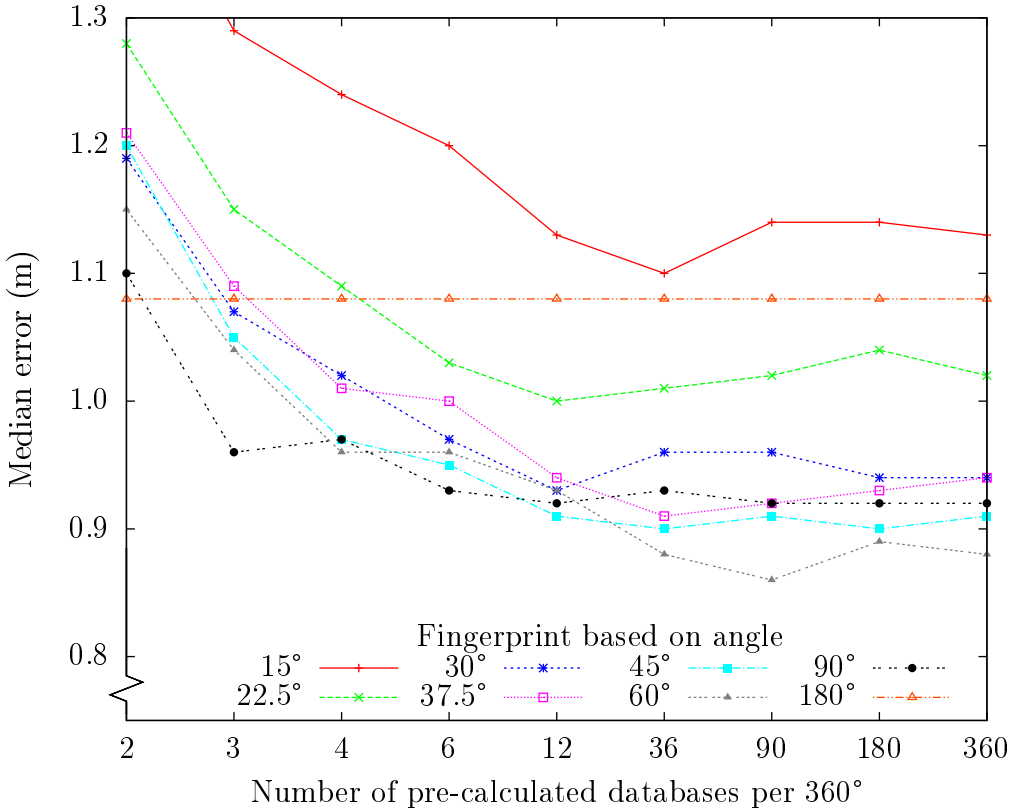



Figure 3.8: Positioning errors using SSD-O, for different values for $\alpha$, and different number of databases.

cially above 36 the change is small, and even with just 3 or 4 databases, performance is better than in the SSD case (the 180° line). How many databases should be pre-calculated depends on the application and the amount of storage space available for the databases.

SSD-O uses the device's compass to determine orientation. During surveying and positioning no obvious errors in the orientation reported by the smartphone's compass were observed, however the compass accuracy was not measured. In other environment the smartphone's compass was seen to give errors that were off by 90°, therefore it is interesting to look at what happens when the compass device gives wrong readings. For figure 3.9 the heading readings during the positioning were offset by different values. The

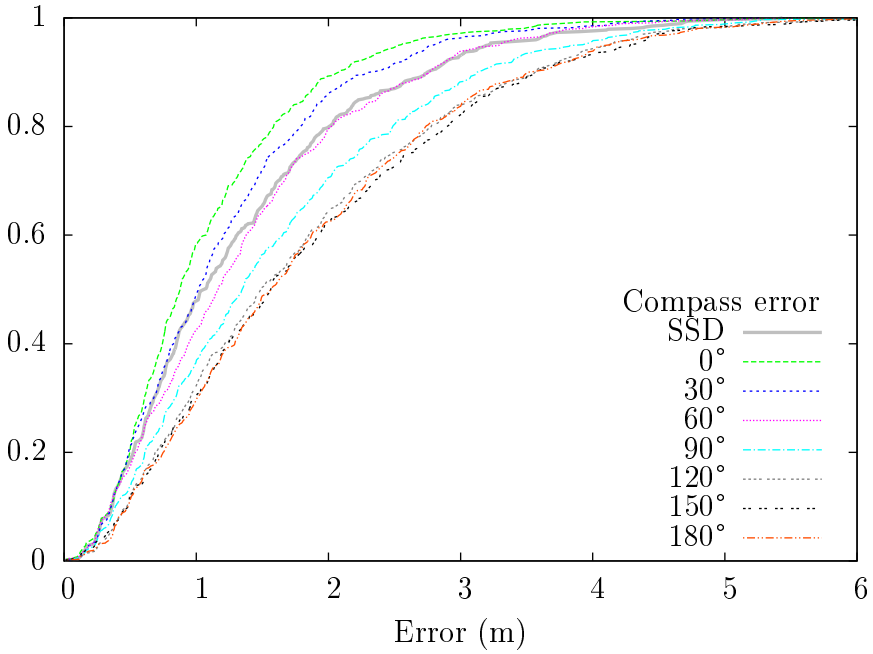



Figure 3.9: Influence of compass error on SSD-O.

figure shows that SSD-O outperforms SSD with errors of less than 30-60°.

### 3.4.4 Blackout Resistant Positioning

Figure 3.6 shows that BRP slightly outperforms SSD, however not by much, and worse than SSD-O. It is still interesting because it gives an alternative positioning method, based on the radio properties of BLE, and because it is more resilient against unobserved beacons, either because of a shorter listening period (section 3.5.3) or because of other reasons (section 3.5.2).

It is also of interest that BRP and SSD have errors in different locations; a combination of the two could be used to further improve positioning, although the way in which to make this combination needs further research. As figure 3.6 shows, taking the average of the locations returned by SSD and BRP (labelled SSD+BRP), already results in a positioning method that is better than either one.

### 3.4.5 Blackout Resistant Positioning with Orientation

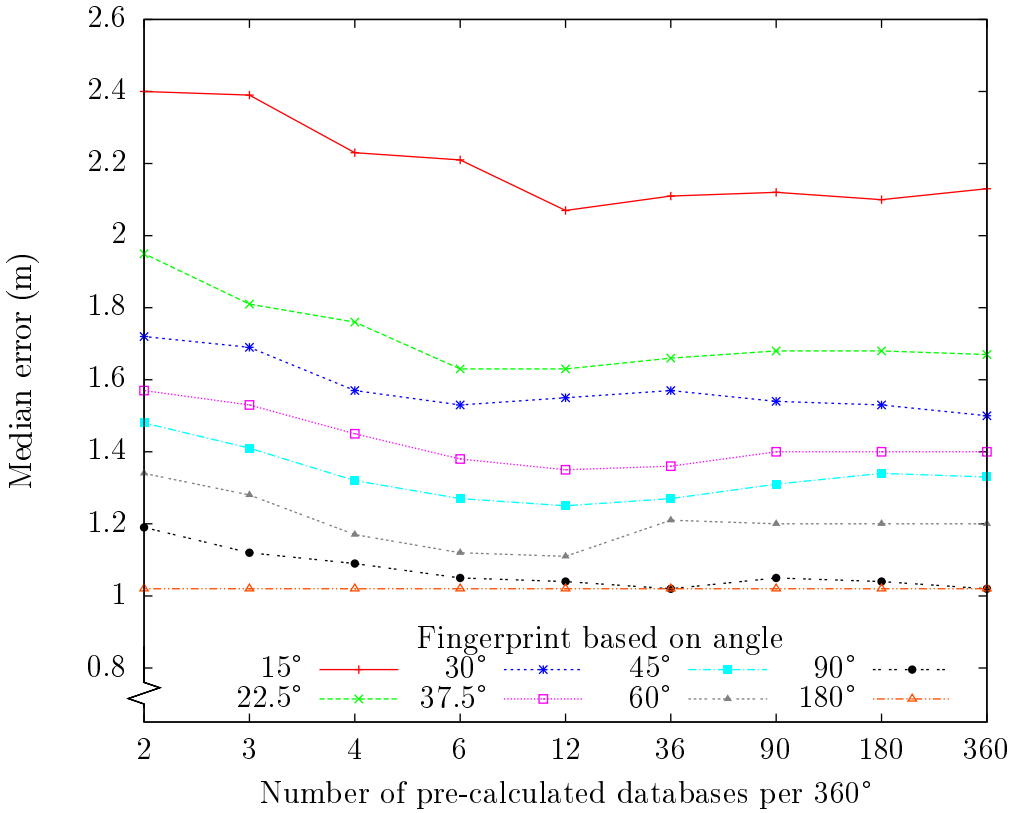



Figure 3.10: Positioning errors using BRP-O, for different values for $\alpha$, and different number of databases.

As expected, BRP-O does not give the same performance benefits over BRP as SSD-O has over SSD. It has the same median performance as BRP, it does however have a lower mean (1.19m vs 1.25m) and 75th percentile (1.58m vs 1.67m). It should be noted however that the BRP penalty function was optimised for the BRP situation, and different parameters are needed for the BRP-O function. In section 3.5.3 I show that BRP-O does outperform all other methods on shorter listening intervals.

I change the same parameters for BRP-O as I did for SSD-O. A larger $\alpha$ gives a smaller error; it is possible that choosing different parameters for

the penalty function results in different optimal $\alpha$. As figure 3.10 shows, reduction of the number of databases to 36 or even 12 or 6 can be done without sacrificing much accuracy; some $\alpha$ lines even seem to increase with fewer databases, I expect this to be by chance though.

Figure 3.6 shows that, although BRP-O performs worse than SSD-O, taking the average of the locations of both (labelled SSD-O+BRP-O), results in better performance, especially for those locations with a larger error.

### 3.4.6 Blackout Resistant Positioning-Radio Propagation Model

BRP-RPM performs twice as bad as SSD and BRP, however the main advantage is that no surveying has to be done, and the database is extremely small; it only needs to contain information on each beacon's location and transmit power. Efficiently encoded this information is 12 bytes[1], so a beacon could include this information in its advertising packet, completely removing the need for a database.

## 3.5 Change parameters

During this section I look at the performance of the different methods when certain parameters in the system are changed. Throughout the section the median error is used for the performance; I have found this to be a good indicator for the performance of the whole system.

In sections 3.5.1, 3.5.2 and 3.5.4 I "randomly" remove beacons and surveying points. In these cases the results are gotten by doing 100 iterations, removing

[1]Assume we want a world-wide coordinate system with a resolution in the cm-range. Earth's surface area is 510,072,000 $km^2$; any usable coordinate is between -10km and +200km in height, giving 107,115,120,000 $km^3$, or $10^{26}cm^3$ to address – any errors due to the earth not being flat fall within error margins. 11 bytes covers $3 \times 10^{26}$ addresses. The RSS value is only 1 byte, hence 12 byte is enough.

the beacons and points in a different order each time. The order in which the beacons and points are removed is determined by sorting md5-hashes of the beacon ID/point coordinates and the iteration number, resulting in a deterministic pseudo-random order. The median error plotted is the average of the median errors of the 100 iterations.

### 3.5.1 Number of beacons

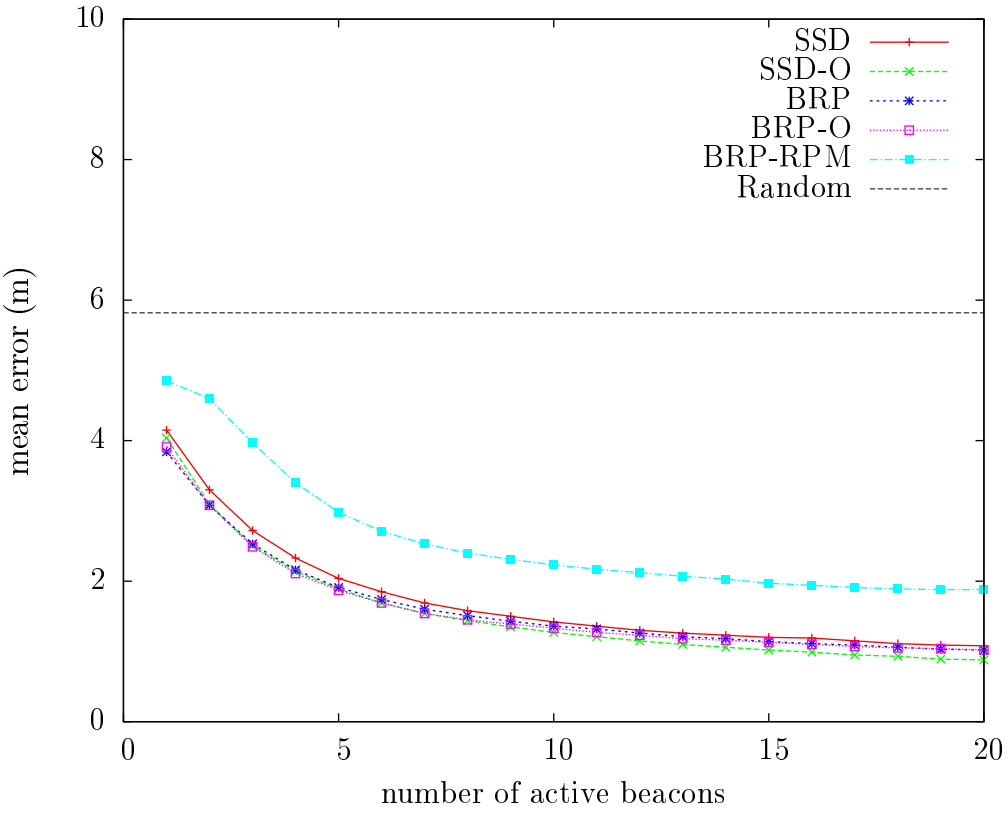



Figure 3.11: Effect of the number of beacons

Every beacon added improves positioning, but results in diminishing return. The test bed contained 20 beacons, about 0.1 per $m^2$. Figure 3.11 shows all methods having gradually increasing errors when fewer beacons are used.

This effect is slightly smaller in the BRP-based methods, and they outperform SSD-O for a small number of beacons.

### 3.5.2 Unobserved beacons

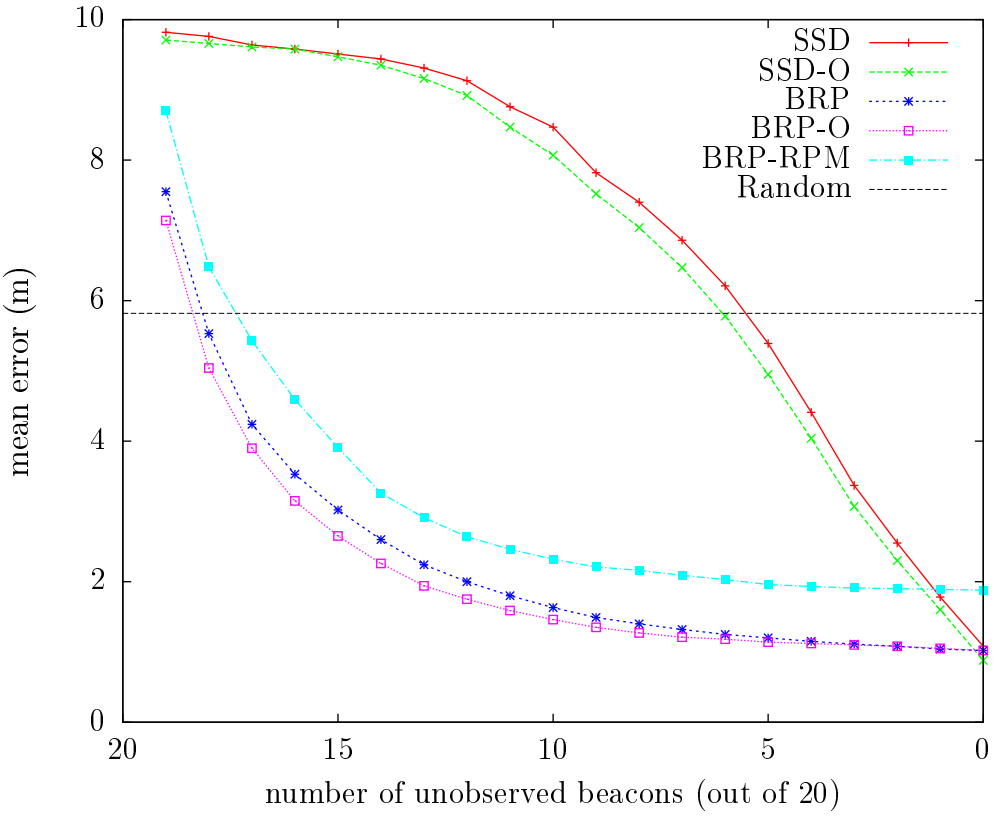



Figure 3.12: Effect of unobserved beacons

In the previous section, I considered what happens if there are fewer beacons, both during surveying and positioning. In this section I explore what happens if the beacons were seen during surveying (hence are present in the fingerprint database), but not during positioning. This may be because a beacon was switched off, the battery died or the beacon was purposefully removed in a denial-of-service attack. Since Wi-Fi access points are plugged in to mains, and they are usually in locations not easily accessible to attackers, they are

less likely to suffer from the latter two failures than BLE beacons.

Another reason that a beacon may be unobserved is that all packets from the beacon during the listening interval were lost. Section 2.5 shows that there is a lot of packet loss, and beacons may not be observed, even after several beaconing intervals. This specific situation is investigated more in section 3.5.3, where the listening intervals are shortened.

If no packets have been received from a beacon during either surveying or positioning, the algorithms used set its average and maximum RSS -105dB.

Figure 3.12 shows that the both SSD based methods are very sensitive to unobserved beacons. A single unobserved beacon almost doubles the mean error for both methods. The BRP methods however are designed to resist blacked-out beacons, and only show gradually increasing errors in a situation where beacons are not observed. Even with a single unobserved beacon, BRP outperforms SSD-O and SSD by having a 35% and 42% smaller median error.

### 3.5.3 Positioning listening length

As described in section 3.2, positioning was done by listening at a single location for 2 seconds. On average, 197 packets were received during one positioning, meaning a 48% packet loss (based on the actual 9.5Hz advertising rate for the beacons). Figure 3.13 shows that only 1.5% of positionings fail to observe all beacons with a 2 second listening interval, but at shorter intervals, more positionings do not observe all beacons.

Section 3.5.2 shows the effect that unobserved beacons have on positioning. In addition to beacons being unobserved, a shorter listening interval also leads to fewer packets for those beacons that are observed, meaning that the calculated mean or maximum RSS is less accurate.

Figure 3.14 shows the effects of a shorter listening period. Decreasing the listening period from 2 seconds, the error increases slightly until about 0.7 seconds, when it sharply increases for the SSD based methods, and less, but

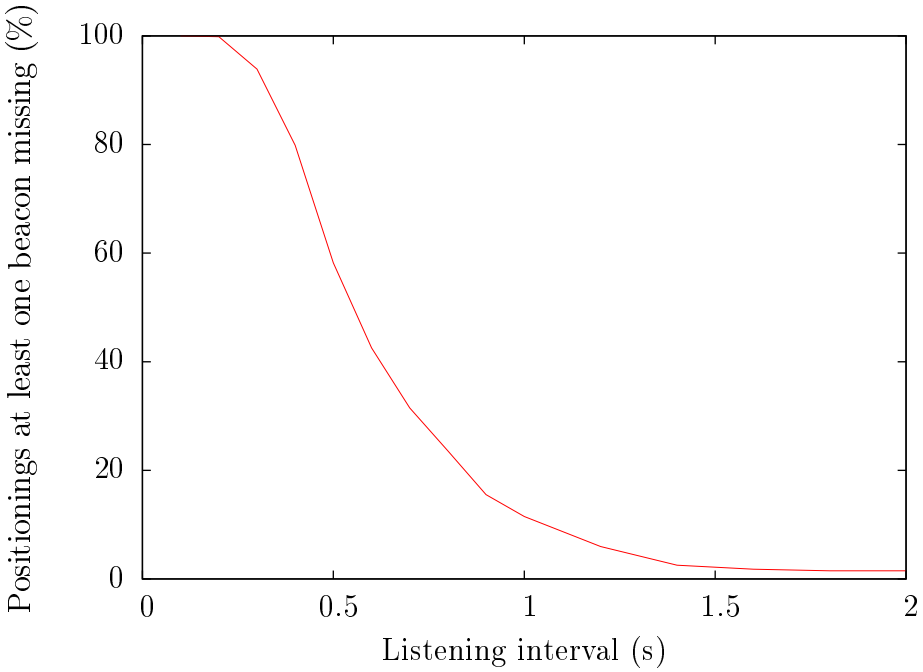



Figure 3.13: Percentage of positionings with unobserved beacons

still considerate, for BRP based methods.

With 30% of positionings having at least one beacon unobserved at 0.7 seconds (figure 3.13), and the knowledge from section 3.5.2, this result is not unexpected.

### 3.5.4 Number of surveyed points

In order to see the effect of the number of points in the fingerprint database, I removed survey points from the fingerprint database, while still using the same data for positioning. The points were removed in the deterministically-random way described before.

As figure 3.15 shows, removing fingerprints reduces accuracy, as one might expect. Reducing the database to 100 points, a 55% decrease, only increases the median error with around 15%, however removing more points results in a stronger increase in median error.

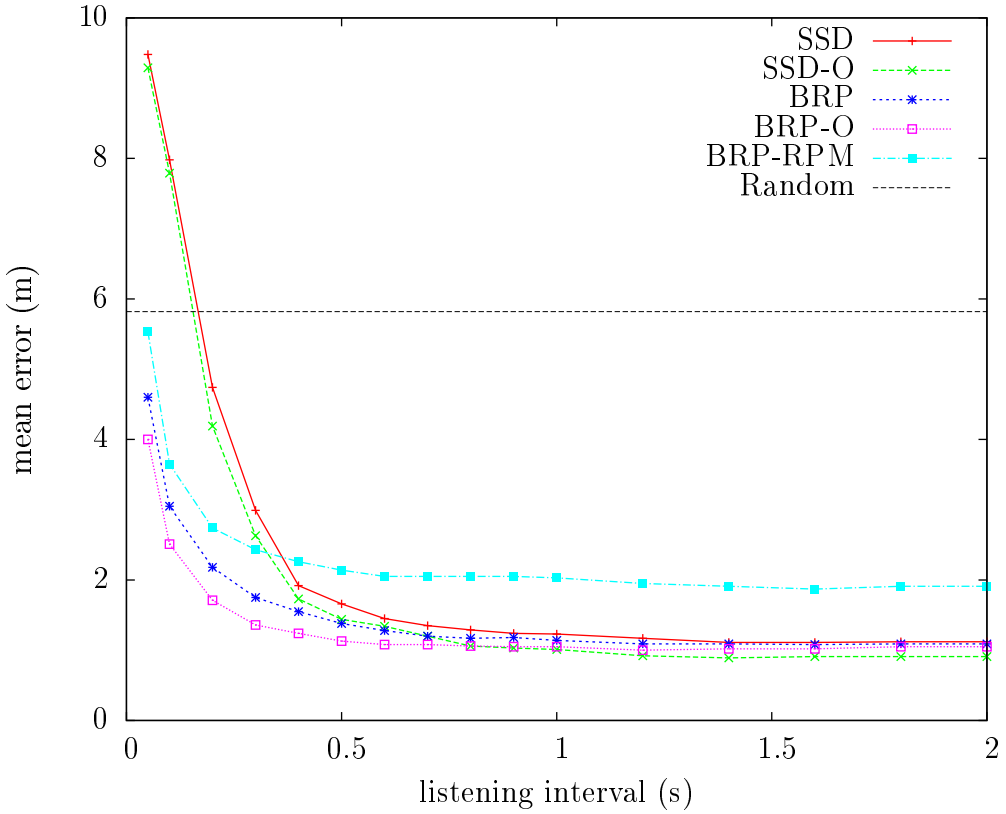



Figure 3.14: Effect of positioning listening interval on median error

The BRP based methods are shown to be more resilient against removal of even more surveying points than the SSD based methods, even though I do not have an explanation for this. The BRP-RPM does not use a fingerprint database, and therefore always performs equally good.

## 3.6 Discussion

In this chapter I demonstrated that push-to-fix positioning using BLE is possible using a beacons and an iPhone, and I introduced some new algorithms and showed they work. It was intended as such, and in no way does it answer all issues in BLE positioning. In this section I discuss some factors that have

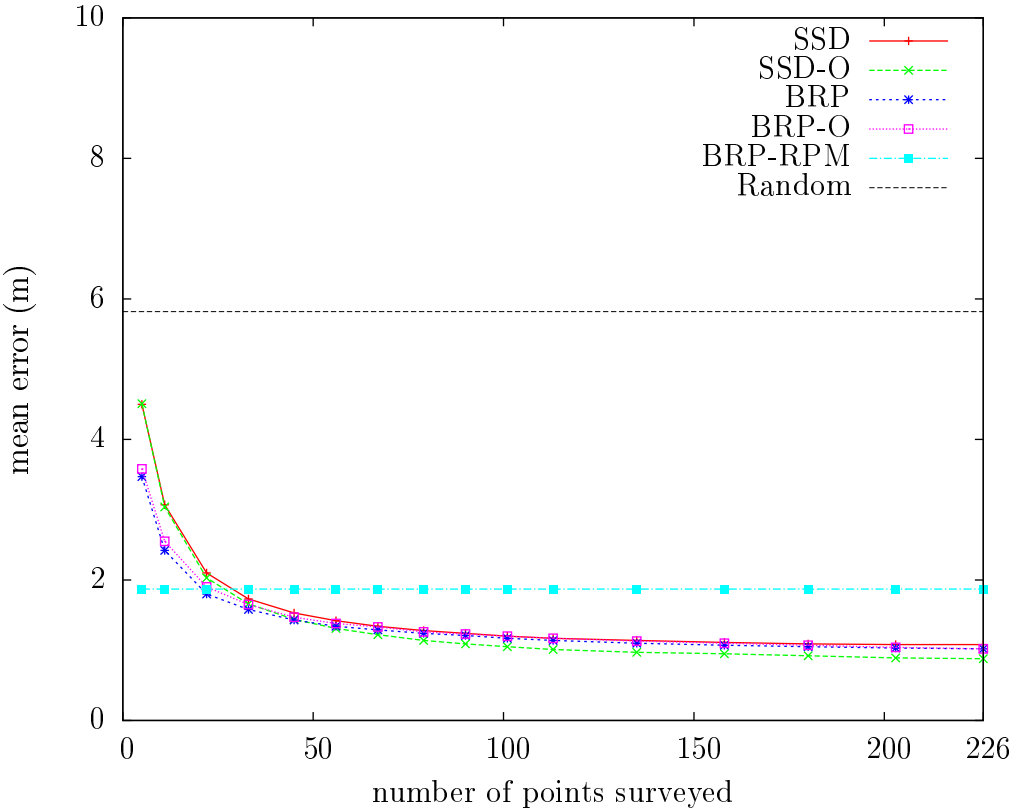



Figure 3.15: Effect of reducing the number of points surveyed

not been taken into account. Each of them lends itself to further research.

The device used to do the surveying and the positioning was the same physical device. Do different devices return different RSS for the same signal, and if so, how can a positioning method correct for that?

The test was done in a 2D environment; applying this to a 3D environment is not trivial. Indoor environments often span multiple floors and beacons can be received through floors and ceilings. A 2 meter error horizontally is probably acceptable in many positioning scenarios, while 2 meter vertically may position you on a different floor.

The test bed environment was mostly deserted during both surveying and positioning. Section 2.5 shows that people moving around results in large

RSS fluctuations in both directions. How do the positioning methods in this report perform in an environment where many people move around.

Finally the test bed used in this report is a relatively open space. Walls may block or weaken BLE signals. Because BRP is designed to be less sensitive to such signal drops, it may "miss" these hints about on which side of a wall it is. I expect BRP to perform as well as SSD in this, provided there are enough beacons on both side of the wall, however further research will have to show this.

## 3.7 Access to the fingerprint database

One question we have not dealt with above is how the smartphone has access to the fingerprint database. Although this does not influence the results of the positioning, it is interesting to consider, especially in the light of chapter 4 on privacy and security. The database size depends on the area that is being mapped, around 20 bytes per $m^2$ gives a fair estimation[2], resulting in about 50 kilobyte for the average Tesco Superstore[3], up to 1 megabyte for large museums such as the British Museum or le Louvre.

Below I suggest several strategies how to access a fingerprint database from a positioning device.

[2]Assume a $10,000m^2$ area that we want to map. I assume a compact data format, which starts with a list of beacons, and then surveyed information, one survey point per meter, 10,000 survey points in total. This results in 10,000 data points, for each I'll store the RSS of the 20 *closest* beacons. Since the coordinates for all beacons are known, and the coordinates for each surveyed point are known, the smartphone can calculate the 20 closest points. The RSS was returned by all test devices as signed 8-bit value.

This means that for the beacon-list, we need 8 bytes per beacon (2 byte beacon ID and 2 bytes per $x$, $y$ and $z$ coordinate), and for the actual RSS $20 \times 1byte$ per position. Assuming a beacon every 5 meters, 400 in total, the map will be $someoverhead + 400 \times 8 + 10,000 \times 20 \times 1 = 200.4kilobyte$, which results in about 20 bytes per $m^2$.

[3]`http://www.tescoplc.com/files/pdf/results/2014/prelim/prelim_2013-14_analyst_pack.pdf`, accessed on 10 June 2014.

### 3.7.1 Global beacon database

Wi-Fi positioning is based on a global database (more precisely multiple competing global databases), maintained by commercial parties such as Skyhook and Google. The data from this database comes from surveying outside (for instance with Google Streetview cars), and is updated by manual submissions and by use[4]. One may try to build something similar for BLE. Firstly it should be noted that one has to be able to distinguish BLE beacons (which are stationary and have a fixed ID) from other BLE peripherals that advertise, such as smart watches or heart rate meters, which are both mobile, and can change their advertising message. As of this moment there is no standard from the Bluetooth SIG which allows one to recognise an unknown device as a BLE beacon from its advertising message, or to govern the beacon IDs so that a single ID is not reused. There are however not-Bluetooth SIG standards out there, such as iBeacon, that could fill this role. Secondly BLE beacons typically have a limited range, and may not be noticed when driving around outside, so surveying inside may be needed. Finally there may be a chicken-and-egg problem: unlike Wi-Fi access points, BLE beacons have only one goal, which is positioning, and they will not be installed until the database to use them is there[5]. The database however only makes sense with a critical mass of BLE beacons.

If these problems were solved (either because companies would manually enter their beacon positions into the database, or through better scanning techniques), and everybody agreed on a standard, a BLE positioning system similar to the Wi-Fi positioning system, could work. Because of its size, downloading the full database would only be feasible in certain situations, and most of the time positioning would be done by sending the measured RSS to the cloud, and getting back a position. Ideally such a database would be open for anyone to use, so that BLE positioning would work in any application.

[4] `http://www.skyhookwireless.com/location-technology/coverage.php`, accessed on 10 June 2014.

[5] Even though beacons can also be used in a stand-alone way, such as iBeacons.

### 3.7.2 In-app beacon database

Instead of having a global database with all BLE fingerprints, this information can be available on the phone itself. Typically the company using the building could provide an app with navigation support for that building (e.g. Tesco releasing a Tesco app, which allows navigation inside Tesco supermarkets). An advantage of this system is that no internet connection is needed to do the positioning. Navigation happens in a proprietary app though, which needs to be installed before. This method also has increased privacy concerns, that I discuss in chapter 4. Finally this method may result in an out-of-date database; Möller et al. [2012] showed that only 50% of Android users install a new app version within 7 days, and 25% has not updated even after 3 months.

### 3.7.3 On-beacon beacon database

A third option is to have the fingerprint database and the indoor map on the beacons themselves, preferably in an open format. Since advertising packets themselves can only contain 31 bytes of information (this includes some data describing the packet as to be from a beacon, and the beacon ID), a phone would have to make a connection to a beacon to download the database and map. In some quick experiments, I found a sustained BLE data transfer rate of 8KByte/second, meaning that a small database and map can be downloaded in seconds[6]. The beacon database and indoor map can be picked up by any app that understands the format, and no internet connection is required. Extra care does have to be taken that all beacons have the same up-to-date version of the database, and because beacons are connected to and send out more data, their battery life will decrease. Depending on how important it is that the database and map can not be spoofed, they may also need to be digitally signed.

[6]Larger maps could possibly be downloaded in parts, or by switching to a higher bandwidth system.

It should be noted that this technology can be extended beyond just mapping, such as giving up-to-date information in maps (which check out registers or theme park rides are not busy, at what time are the penguins being fed in this zoo, where can I find the next bus to Cambridge, etc), or to replace signs (menu-of-the-day, wifi-password, opening hours or latest offers).

## 3.8 Alternative methods

This chapter focussed on building an opportunistic RSS based positioning system with BLE beacons. In this section I discuss two alternatives to this system, also based on BLE and standard smartphone hardware, that I think are worth some further research as well.

### 3.8.1 Bluetooth Low Energy Bats and Crickets

Positioning systems *Active Bat* [Harter et al., 2002] and *Cricket* [Priyantha et al., 2000] use a combination of radio waves and ultrasound to do positioning; using the difference in speed between radio and sound waves, the distance to multiple beacons can be determined, from which the position can be calculated. Active Bat and Cricket require the users to carry a specific device in order to use the system.

With BLE, all the technology needed for such a system is available in a smartphone: a speaker/microphone, and a way for apps to send and receive radio packets.

Multiple combinations of what device sends what signal are possible, but a simple set-up, similar to the one tested above, could have beacons send both a BLE advertising packet and a near-ultrasound signal. The BLE packet is picked up by the smartphone, which then listens for the near-ultrasound signal. There is evidence that smartphone microphones can pick-up (near) ultrasound[Arentz and Bandara, 2011, Bihler et al., 2011], and in a short test the iPhone 5S picked up sounds to 20kHz without a problem, with higher

frequencies not tested. For this to work it is essential that the information an app receives on when a BLE packet was received is accurate enough.

### 3.8.2 Listening beacons

Instead of having the smartphone listen for packets from the beacons, the smartphone instead could send advertising packets to be picked up by the beacons. The beacons could then use all sorts of methods (for instance Time-Difference-Of-Arrival) to determine the sender's location, and send this back. Using something else than RSS for positioning, means that the beacons may need to be more complex, probably making them more expensive, but this may be acceptable in some deployments. The beacons will need to collaborate to determine the smartphone's position, so they need to be networked, and since they need to listen all the time, they will use more energy than the simple beacons we used above.

Having the smartphone send signals instead of just listening for them, has privacy implications. The next chapter discusses the privacy and security side of positioning methods.

# Chapter 4

# Privacy, security and performance

## 4.1 Introduction

In the previous chapters I showed the technical feasibility of a push-to-fix positioning system based on BLE. As noted in the introduction, another requirement for such a system is social acceptance. When location data was found to be collected on iPhones in 2011[1], presumably violating user's privacy, this set of an worried reaction from the public[2]. Recent realisation that one's phone can be tracked by shop owners, has led to its own angry reactions[3]. Since recent revelations on government data collection have most likely only strengthened the public's desire for privacy.

In this chapter I compare the privacy, security and performance of BLE based positioning systems to that of some well-known alternatives. I score each method on 12 points in three categories: $+$, $-$ and $\pm$ The system is set up so that $+$ is always more desirable than $\pm$, which in turn is more desirable than $-$. Many of the scores are debatable; where needed I give my

[1] `http://radar.oreilly.com/2011/04/apple-location-tracking.html`, accessed on 12 June 2014.

[2] `http://abcnews.go.com/Technology/apple-pushed-congress-answers-iphone-tracking/story?id=13426917`, accessed on 12 June 2014.

[3] `http://www.nytimes.com/2013/07/15/business/attention-shopper-stores-are-tracking-your-cell.html`, accessed on 12 June 2014.

arguments in the text. I feel I scored the methods objectively, but someone else may come to other results on the same points. The produced score card should therefore not be taken as a means to come to a single, final, score for each method; it does give a quick overview of some of the methods in use, and their strong and weak points.

## 4.2 Methods

The methods are divided in 4 categories: Non-technological methods are using a paper map and street signs, and asking for directions. In traditional technological methods I distinguish traditional GPS, assisted GPS (A-GPS; GPS augmented with information from an internet connection), Wi-Fi positioning. I further consider RSS-based BLE positioning, such as discussed in the previous chapter, in combination with each of the fingerprint-database access methods from section 3.7, and finally the two alterative positioning methods from section 3.8.

Many methods can be implemented in different ways; if there is a prevalent default way, I look at that one, else I assume the way that scores best.

## 4.3 Scoring

### 4.3.1 Privacy

Privacy is the ability to do positioning without anybody else finding out that you are doing so. In addition to the party for whom the positioning information is meant, there are three parties possibly involved. The *beacon owner* is the entity that manages the beacon-network. The *database owner* is the owner of the global database, for systems that use them. *Third-party* is an arbitrary eaves-dropper; it implies some large-scale automatic way of doing this, not just standing somewhere and watching. A second, more serious,

| | | Paper map and street signs | Ask for directions | GPS | A-GPS | Wi-Fi | BLE Global database | BLE in-app database | BLE in-beacon database | BLE Bats & Crickets | Listening beacons |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Privacy | beacon owner | + | ± | + | + | + | + | − | + | + | − |
| | database owner | + | + | + | ± | − | − | + | + | + | + |
| | third party | + | + | + | + | + | + | + | + | + | − |
| | untackable | + | ± | + | + | + | + | − | + | + | + |
| Security | passive DOS | ± | + | + | + | ± | − | − | − | − | ± |
| | active DOS | ± | + | − | − | − | − | − | − | − | − |
| | passive spoof | ± | + | + | + | ± | − | − | − | − | ± |
| | active spoof | ± | ± | − | − | − | − | − | − | − | − |
| Performance | time to fix | ± | ± | − | + | ± | + | + | ± | + | + |
| | accuracy | ± | ± | ± | ± | − | ± | ± | ± | + | + |
| | scalability | + | ± | + | + | + | + | + | ± | + | ± |
| | affordable | ± | + | − | − | ± | ± | + | + | + | ± |
| | works indoors | − | + | − | − | + | + | + | + | + | + |
| | no internet | + | + | + | − | − | − | + | + | + | + |
| | up-to-date | − | ± | + | + | + | + | − | + | + | + |

Table 4.1: Scores for different positioning methods

way in which privacy can be lost is if two positionings done (either in short succession or days apart) can be determined to be from the same smartphone; if not I determine the method to be *untrackable*.

The paper map scores + on all four points: even though somebody may see you using the map, and recognise you next time, this is the same in all methods and not considered. Asking for directions scores + on third party and database owner, however only ± on beacon owner (in this context the person asked is the beacon) and untrackable, since the person used for getting the directions, has information on your position and may recognise you next time.

GPS is a technique that does not require any signal from the phone to a central authority, so scores + on all privacy issues. A-GPS on the other hand can be used in two ways: in Mobile Station Based mode, only global information is read from the internet, no revealing privacy sensitive data; in Mobile Station Assisted mode positioning data is sent to a central location, resulting in a ± for database owner. I assume, without having confirmed, that in this latter case no information that could be used to track the device over requests is being sent, resulting in a + on untrackable. Using Wi-Fi positioning, a database owner learns the fingerprint query made (−), but no other privacy sensitive data is shared.

BLE with a global database has equal privacy performance to Wi-Fi. An in-app database has different characteristics. Since the app used to do the positioning is owned by the owner of the beacon network, the owner has access to both the position of the phone, and can identify the same phone again next time, so on both owner and untackable this method scores −. BLE with an in-beacon database does not need a third party, and downloads only "global" information from a beacon, giving the method two + s.

Bats & Crickets score perfect on privacy, provided that the beacons broadcast their own position in the same way that was suggested for BRP-RPM beacons in section 3.4.6.

Listening beacons do not use a global database(+), however the owner of

the network knows the location of the smartphone (−). Whether subsequent requests are untrackable depends on how the position is relayed back to the phone, but it is possible to do this in a way that is secure and untrackable. However a third party could place his own positioning listening beacons (e.g. in the shop next-door) and learn the smartphone's location that way.

### 4.3.2 Security

The security score illustrates how resilient the technique is against an attacker. Two types of attacks can happen: either a denial-of-service, where the attacker makes sure that positioning fails, or a spoofing attack, where the attacker aims to let the positioning result in a different position than the actual position. Either type of attack can happen in two ways: active, where the attacker has to be present, or has to have a device present, to succeed, or passive, which only requires that the attacker did something in the past. If a technique is susceptible to a passive attack, this implies that it is at least as susceptible to an active attack. The score on this section depends not only on whether an attack is theoretically possible, but also on whether it is practically feasible; for instance shooting a GPS satellite out of the sky is not considered feasible.

The paper map and street signs attack can be DOS'ed by an attacker removing the street signs, or spoofed by replacing street signs by fake ones. This is a passive attack, however the practical feasibility is limited, and possibly a user will be able to navigate with just the map, even without street signs. This technique therefore scores ± on all attacks. Asking for directions is resilient to most attacks, however a motivated attacker could stake out a location and give a user wrong directions when asked. Since this last one is doubtful to succeed, the technique scores ± on that one, and + on all others.

I do not consider destroying or moving satellites feasible, making both GPS and A-GPS resilient to passive attacks. GPS signals can be jammed[Grant et al., 2009], or spoofed[Tippenhauer et al., 2011], meaning that the techniques are not secure against active attacks. It should be mentioned that

none of the other positioning systems are resilient against active attacks, since signals can always be jammed or listened to and retransmitted on another location. Wi-Fi positioning scores a ± for passive attacks, since the Wi-Fi access points are usually not in a location where an attacker can disable or move them.

Since BLE positioning beacons will typically be placed in more accessible locations than Wi-Fi (because of the shorter range and the lower cost of the beacons), any BLE beacon technique will be vulnerable to passive attacks, such as stealing or moving of beacons, as well. This is equally true for Bats & Crickets. The listening beacons get a ± for these kinds of attacks, since they are networked and will be able to notice disruptions and possibly take appropriate corrective action.

### 4.3.3 Performance

Finally I look at how useful the systems are. *Time to (first) fix (TTFF)* describes how long it takes to get the first fix. Less than 2 seconds scores +, less than a minute scores ±, more than a minute is −. *Accuracy* scores a + if the position reported is within one meter of the true position, ± if it is within 5 meters, − means more than 5 meter. *Scalability* means whether a system can cope with many clients positioning at the same time. *Affordable* talks about the cost of creating and managing the system, notably *not* the cost of doing a single positioning. Whether the system can be used indoors or without an internet connection being present is noted in the next two properties *works indoors* and *no internet*. The final property, *up-to-date* illustrates the ability of the system to quickly incorporate changes.

For paper map or asking directions, I estimate to have an answer within a minute, accurate to within a couple of meters. Both systems work without internet, but where the asking directions is cheap and works indoors, paper maps and street signs only work outdoors, and require purchase of maps and maintenance of street signs. Paper maps are scalable, however if thousands of people were to ask directions every day, very few would be happy to answer.

Maps are not up-to-date, while the knowledge of a person asked for directions may be up-to-date.

GPS and A-GPS differ mostly in the time to first fix (this being much shorter with A-GPS, seconds, as opposed to GPS's 12.5 minutes), and the need for internet (only A-GPS). Accuracy may be slightly better with A-GPS (depending on the mode), but still not within a meter, while scalability is slightly worse for A-GPS, however not enough not to give it a $-$. Both systems are extremely expensive (not necessarily per user, but to launch and maintain in absolute cost) and do not work indoors since GPS signals do not penetrate buildings. Wi-Fi has a $\pm$ time to first fix, since scanning the Wi-Fi channels takes more than 2 seconds on newer phones that support 5GHz Wi-Fi. The accuracy is generally $-$, but may be better in area of high access point density. The system scales well and it costs a bit to maintain the central database. It does work indoors and does not work without internet.

The BLE methods all give a quick time to first fix, except for in-beacon database, which needs time to download the database from the beacon, scoring $\pm$. The accuracy of BLE positioning is a topic of active research (to which this report hopefully makes a contribution). It is not per-se better than Wi-Fi, but since beacons are cheap and can be deployed in great numbers, I score this $\pm$. The in-beacon database may have problems scaling, since each initial positioning occupies a beacon for a while, to download the database and map, scoring $\pm$ on scaling, where the two other BLE technologies scale fine. The central database costs some money to set-up and maintain, the other two BLE technologies are cheap. All three work indoors, and only the central database needs an internet connection. Finally, the update problems with an in-app database have been discussed in section 3.7.2; both other BLE methods can receive updates easily.

Bats & Crickets score very well on all points, however accuracy greatly depends on how accurate the time stamps on BLE packets are.

Finally the listening beacons technology has a fast time to first fix, (possibly) high accuracy, however the technology for this high accuracy will make the

beacons more expensive. Since one or more beacons are occupied to do the positioning, the systems scores $\pm$ on scaling. It works indoors, and does not need an internet connection.

# Chapter 5

# Conclusions and further research

## 5.1 Conclusions

In this report I show that a positioning system based on BLE beacons is possible, and allows in the test-set-up for positioning with a median error of less than one meter.

### 5.1.1 Research question 1

*What are the radio propagation properties one has to take into account when trying to build a push-to-fix Received Signal Strength (RSS) based positioning system based on Bluetooth Low Energy?*

The RSS for a BLE beacon does not decrease smoothly with distance to a beacon, something which would be ideal for positioning. I have identified several of the factors responsible for this. Section 2.3 showed that multi-path interference results in drops of more than 20dB in locations only several centimetres wide. Each of the three advertising channels shows these drops in different locations.

In section 2.4 I have shown that the orientation of the smartphone in relation to the transmitter has a large influence; sometimes turning the smartphone

just 15° results in a 15dB RSS drop. Furthermore a human body between the transmitter and smartphone also has an influence on the RSS.

Section 2.5 showed how people moving through a room influence the RSS. The RSS of beacons in a busy room fluctuates as people move around; in some situations packets had a 20dB higher or lower RSS in the room with people than in the same room when it was empty.

As shown in section 2.6, not each BLE packet was received by the test equipment. For a 1Hz beacon, outside in a field, only 76% of packets were received, while in tests with 20 beacons at 10Hz in a busy environment, this dropped to 41%.

### 5.1.2 Research question 2

*Can we use the positioning methods that were developed for Wi-Fi positioning for BLE positioning? Can I find an RSS-based push-to-fix positioning algorithm that works better than these, by taking into account some of the unique properties of BLE, mentioned in the last question.*

In chapter 3 I have shown how well a number of algorithms work in BLE positioning, in a room of 18 by 13 meters with 20 beacons, each transmitting at 10Hz. After creating a fingerprint database with 226 points, I did positionings on 670 locations, listening for BLE packets for 2 seconds on each. The signal space distance (SSD) method, a popular positioning method used in Wi-Fi positioning, has an error smaller than 3.22m in 95% of the cases. A variant of this method which takes direction into account, SSD-O, gave an 18% smaller error on average, scoring 2.55m on the 95th percentile. In section 3.3.4 I introduced Blackout Resistant Positioning (BRP), a positioning method that takes specific radio propagation properties of Bluetooth Low Energy into consideration, and a variant which also takes orientation into consideration, BRP-O. BRP performed slightly better than SSD, while BRP-O performed worse than SSD-O, with a 95th percentile error of 3.10m and 2.73m respectively. A combination of SSD and BRP, where the average

position of both methods is taken, scored 3.06m as 95th percentile, while a similar combination of SSD-O and BRP-O scored 2.40m.

If a shorter listening interval is taken (see section 3.5.3), the BRP (-O) methods performed much better than the SSD(-O) methods, showing a 95th percentile error of 4.63m and 3.64m for BRP and BRP-O against 5.38m and 4.72m for SSD and SSD-O, when listening for 0.5 seconds.

### 5.1.3 Research question 3

*How does a BLE based positioning method compare on security and privacy to other positioning methods? What suggestions can be made to enhance privacy and security?*

In chapter 4 I discussed the privacy and security properties of several positioning systems, both BLE and not. The BLE positioning method experimented with in this report is an opportunistic method, meaning that positioning is being done solely on information received, without transmitting. Therefore the positioning method itself can be done in perfect privacy. However the method depends on a fingerprint database, which has to be obtained. A fingerprint database in the cloud has the same privacy as Wi-Fi positioning; an in-app database is unable to give privacy guarantees, while a database on the beacons themselves gives a lot of privacy guarantees.

Since BLE beacons will typically be installed close to the users, they are on easily accessible spots, and therefore vulnerable to people, either maliciously or by accident, moving or removing them.

## 5.2 Further research

Push-to-fix positioning using Bluetooth Low Energy is a new field, and this report is only a first quick survey of the possibilities and problems. The work in this report can be extended or augmented in many directions. Some

suggestions for further research are already mentioned in the text, especially in section 3.6, more are listed below.

- Section 2.5 shows that RSS fluctuates a lot when people are moving around the room. How do the positioning methods discussed in this report perform under those conditions?
- All measurements in this report focussing on packet loss were done with iOS devices, which all showed the similar packet loss. Packet loss influences the accuracy for positioning. How high is the packet loss on other smartphones?
- I chose in my experiments to use an RSS of -105dB if a beacon was not observed. In light of the high packet loss, perhaps another choice, either another RSS or a completely different way of dealing with this issue, is more appropriate.
- BRP does not use information about on which channel a packet was received. The idea behind BRP however may work for individual channels as well as for the RSS of all three channels combined. Does a Blackout Resistant Positioning version that looks at each channel individually have improved performance?
- Section 3.4 shows that the average of the position returned by SSD-O and BRP-O gives a better result than either one alone. Another combination of the two methods may yield better results.
- In section 3.8.1 I suggest an alternative BLE based positioning method. Exploring its feasibility would be very interesting. As a first question to answer is how accurate the timestamp of reception of a BLE packet is for different smartphones.

# Bibliography


Will Archer Arentz and Udana Bandara. Near ultrasonic directional data transfer for modern smartphones. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 481–482. ACM, 2011.

Paramvir Bahl and Venkata N Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. Ieee, 2000.

Roberto Battiti, Nhat Thang Le, and Alessandro Villani. Location-aware computing: a neural network model for determining location in wireless lans. 2002.

Pascal Bihler, Paul Imhoff, and Armin B Cremers. Smartguide–a smartphone museum guide with ultrasound control. *Procedia Computer Science*, 5:586–592, 2011.

Bluetooth SIG. Core specification 4.0. Technical report, Bluetooth SIG, June 2010. https://www.bluetooth.org/en-us/specification/adopted-specifications.

Paul Castro, Patrick Chiu, Ted Kremenek, and Richard Muntz. A probabilistic room location service for wireless networked environments. In *Ubicomp 2001: Ubiquitous Computing*, pages 18–34. Springer, 2001.

Brian Ferris, Dieter Fox, and Neil D Lawrence. Wifi-slam using gaussian process latent variable models. In *IJCAI*, volume 7, pages 2480–2485, 2007.

Alan Grant, Paul Williams, Nick Ward, and Sally Basker. Gps jamming and the impact on maritime navigation. *Journal of Navigation*, 62(02): 173–187, 2009.

Andy Harter, Andy Hopper, Pete Steggles, Andy Ward, and Paul Webster. The anatomy of a context-aware application. *Wireless Networks*, 8(2/3): 187–197, 2002.

Robin Heydon. *Bluetooth low energy: the developer's handbook*. Prentice Hall, 2013.

Kamol Kaemarungsi. Efficient design of indoor positioning systems based on location fingerprinting. In *Wireless Networks, Communications and Mobile Computing, 2005 International Conference on*, volume 1, pages 181–186. IEEE, 2005.

Thomas King, Stephan Kopf, Thomas Haenselmann, Christian Lubberger, and Wolfgang Effelsberg. Compass: A probabilistic indoor positioning system based on 802.11 and digital compasses. In *Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization*, pages 34–40. ACM, 2006.

Binghao Li, Yufei Wang, Hyung Keun Lee, Andrew Dempster, and Chris Rizos. Method for yielding a database of location fingerprints in wlan. *IEE Proceedings-Communications*, 152(5):580–586, 2005.

Andreas Möller, Florian Michahelles, Stefan Diewald, Luis Roalter, and Matthias Kranz. Update behavior in app markets and security implications: A case study in google play. In *Proc. of the 3rd Intl. Workshop on Research in the Large. Held in Conjunction with Mobile HCI*, pages 3–6, 2012.

Dhruv Pandya, Ravi Jain, and Einil Lupu. Indoor location estimation using multiple wireless technologies. In *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on*, volume 3, pages 2208–2212. IEEE, 2003.

Nissanka B Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 32–43. ACM, 2000.

Beomju Shin, Jung Ho Lee, Taikjin Lee, and Hyung Seok Kim. Enhanced weighted k-nearest neighbor algorithm for indoor wi-fi positioning systems. In *Computing Technology and Information Management (ICCM), 2012 8th International Conference on*, volume 2, pages 574–577. IEEE, 2012.

Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. On the requirements for successful gps spoofing attacks. In

*Proceedings of the 18th ACM conference on Computer and communications security*, pages 75–86. ACM, 2011.

Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, 1992.

Paul A Zandbergen. Accuracy of iphone locations: A comparison of assisted gps, wifi and cellular positioning. *Transactions in GIS*, 13(s1):5–25, 2009.