Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Civil and Environmental Engineering

Christopher Wennersteen Gjøvåg

WiFi RTT for Indoor Localization using Google WiFi and Google Pixel 3a

Master's thesis in Ingeniørvitenskap og IKT

Supervisor: Terje Midtbø

July 2020

NTNU
Norwegian University of Science and Technology

NTNU
Faculty of Engineering Science and Technology
Department of Civil and Environmental Engineering

Master thesis

(TBA4925 - Geomatics, Master thesis)

Spring 2020

for

Christopher Wennersteen Gjøvåg

WiFi RTT for indoor localisation using Google WiFi and Google Pixel 3a

BACKGROUND

Using existing WiFi infrastructure has long been considered a cost effective way of determining indoor location. This has normally been done using Received Signal Strength (RSS), however, this method may have limited accuracy. Therefore, the new standard 802.11mc introduced WiFi Round Trip Time (RTT). This uses Time-of-Flight (ToF), and promises to be more accurate.

TASK DESCRIPTION

Give an overview of some of the existing methods for indoor localisation, as well as more specifically how WiFi RTT works compared to these. Then an experiment should be conducted specifically investigating the precision and accuracy of a specific implementation of the standard using the Google Wifi Access Points (AP) and Google Pixel 3a smartphone

Specific tasks:

- Study literature relating to various methods for indoor localisation
- Study literature relating to WiFi RTT specifically
- Develop a simple Android app using the Android WiFi RTT API
- Use data from application to do statistical analysis
- Evaluate the position precision and accuracy of the system

ADMINISTRATIVE/GUIDANCE

The work on the Master Thesis starts on January 15h, 2020

The thesis report as described above shall be submitted digitally in INSPERA at the latest at July 1st, 2020. (Extended from original June 11th deadline)

Supervisors at NTNU and professor in charge:
Terje Midtbø

Trondheim, February, 2020

Abstract

With the introduction of Fine Timing Measurement (FTM) protocol in IEEE 802.11, WiFi now has a standardized way of estimating distance between an Access Point (AP) and a client device that should provide 1 meter accuracy. While previous research into indoor localization has achieved this accuracy and beyond, the ubiquity of WiFi could significantly reduce the friction to indoor positioning.

Previous studies have found varying evidence for the 1 meter accuracy claim. This thesis uses a Google Pixel 3a smartphone and 3 Google WiFi APs to attempt to verify the accuracy using the network of indoor control points in the building of the Department of Civil and Environmental Engineering. It also provides substantial background on previous work on indoor positioning.

The study does not find compelling evidence for the 1 meter accuracy claim using the minimal setup described. Median ranging accuracy is 2 meters, with 90th percentile of 6 meters. The localization error is 6.7 meters median, and 90th percentile of 10.8

Sammendrag

Som følge av introduksjonen av den såkalte Fine Timing Measurement (FTM) protokollen i IEEE 802.11 har WiFi nå en standardisert måte å estimere distansen mellom en trådløs basestasjon og en nettverksklient. Denne protokollen har en påstått nøyaktighet på 1 meter. Selv om tidligere forskning på innendørs posisjonering har oppnådd både denne nøyaktigheten og bedre, har den vidstrakte bruken av WiFi potensiale til å gjøre posisjonering betydelig enklere.

Tidligere studier har funnet varierende bevis for påstanden om 1 meters nøyaktighet. Denne oppgaven bruker en Google Pixel 3a smarttelefon sammen med 3 Google WiFi basestasjoner for å forsøke å verifisere nøyaktigheten. Dette gjøres ved hjelp av fastmerkenettverket som finnes inne i bygningen til Institutt for Bygg og Miljøteknikk (IBM). Den presenterer også vesentlig bakgrunnsinformasjon på tidligere studier om inndendørsposisjonering.

Oppgaven klarer ikke å finne gode bevis for påstanden om 1 meters nøyaktighet. Ved estimering av distanse er den observerte feilen på opp til 6 meter i 90% av tilfellene, med en median på 2 meter. Resultatene for posisjonering er 10,8 meter feil i 90% av tilfellene, med en median 6,7 meter.

Forord

Denne masteroppgaven er utført som en del av studiet Ingeniørvitenskap og IKT, ved Institutt for Bygg og Miljøteknikk, NTNU. Den er også utført mens verden stod ovenfor en fullskala pandemi. Akkurat den siste delen kan jeg ikke si at jeg hadde sett for meg når jeg begynte på NTNU for 5 år siden.

Jeg vil bruke denne anledningen til å takke min veileder Terje Midtbø for hans hjelp og Institutt for Bygg og Miljøteknikk for å ha gitt meg tilgang til campus på tross av situasjonen. Jeg vil også takke mine medstudenter for deres støtte og hjelp i løpet av studiet. En spesiell takk til Magnus Leikvoll for hans evne til å tolerere mitt mas, på tross av at han hadde en oppgave å skrive selv.

Til slutt vil jeg utstede en stor takk til min samboer, Charlotte Sjøberg Sørhus, for hennes tålmodighet og støtte, og til familien min, som alltid har hatt en enorm tiltro til meg.

Christopher Wennersteen Gjøvåg
1. Juli 2020
Trondheim

Contents

Figures

Tables

Listings

Abbreviations

AP	Access Point
CPU	Central Processing unit
CTS	Clear To Send
DBL	Device Based Localization
FTM	Fine Time Measurement
IR	Infrared
ISM	Industrial, Scientific, Medical
I-DBL	Indirect-Device Based Localization
I-MBL	Indirect-Monitor Based Localization
kNN	k-Nearest Neighbours
LOS	Line of Sight
LSM	Least Squares Method
MBL	Monitor Based Localization
NLOS	Non-Line of Sight
RF	Radio Frequency
RFID	Radio Frequency Identification
RN	Reference Node
RTT	Round Trip Time
RSSI	Received Signal Strength Indicator
RTS	Request To Send
TDoA	Time Difference of Arrival
ToA	Time of Arrival
ToF	Time of Flight
TSC	Time Stamp Counter
TSF	Time Synchronisation Function
UWB	Ultra Wide Band

Chapter 1

Introduction

Since their introduction, beginning with civilian use of GPS in the 1980s, GNSS systems have become the de facto standard way of determining user position. A network of satellites allows users to determine their location anywhere in the world, with high accuracy and ease of use. This, combined with the rise of the smartphone, has paved the way for a number of location aware services, ie. services where the location of the user is core to its functionality. A good example is the recent rise in electric scooter rental services. However, one well known limitation of GNSS systems is that they require a Line of Sight (LOS) between the satellite and the receiver. This means that their accuracy is limited in urban canyons, where signals are blocked by tall buildings and skyscrapers, as well as in the indoor environment. This is because the satellite signal does not have the ability to penetrate solid materials.

At the same time, people are spending a lot of their time indoors, in buildings that are both large and complex. Being able to determine ones location in it represents a clear benefit for the end user and, as described by Chen and Clarke (2019), will give rise to new kinds of applications and value layers. To this end, a number of technologies and techniques have been investigated in scientific literature, for use in indoor positioning. Furthermore, a number of complete Indoor Positioning Systems (IPS) have been proposed and developed.

One of the more recent advances is the addition of the Fine Time Measurement (FTM) protocol to the IEEE 802.11 standard. This presents a standardized way to perform localization using WiFi, considered a prime candidate for IPSs due to its ubiquitousness, and promises meter level accuracy according to the WiFi Alliance (2017). On the Android smartphone platform, measurements based on this standard is available to any application running on compatible hardware as of Android 9 Pie.

The goal of this thesis is to: 1. Provide background on the various technologies, techniques and systems presented in literature, and 2. Attempt to verify the claim of meter level accuracy in FTM, using a Pixel 3a smartphone and Google WiFi Access Points (AP), which both provide support for the standard.

The thesis is structured as follows: Chapter 2 provides background, including the FTM standard. Chapter 3 describes experimental setup and execution. Chapter 4 presents experimental results as well as discussions. Finally, chapter 5 provides concluding remarks.

Chapter 2

Background

Section 2.1 and 2.2 will present techniques and technologies used in IPS, while the latter will also mention specific systems. To this end they rely on the survey papers presented by Liu et al. (2007), Yanying et al. (2009), Makki et al. (2015), and Zafari et al. (2019). These all present various IPSs, as well background information on how they work. Then, section 2.3 will present in more detail, preliminary work on time based ranging in WiFi. Finally, section 2.4 will discuss work similar to this thesis.

2.1 Localization Techniques

There are a few different techniques for determining location that has been used in the indoor environment, whether those are distance based or matches some observed variable to a pre-measured one. What follows is an introduction to those methods that has been suggested in literature.

2.1.1 Triangulateration

Many techniques are based on the principle of triangulateration, that is determining the relative position of some unknown point based on the properties of triangles. This can be achieved either by estimating the distances, called trilateration, or by estimating angles, called triangulation. Several different such methods have been suggested for indoor use.

Time of Flight One possible method is observing the propagation time of a signal for a number of known Reference Nodes (RN), known as Time of Flight (ToF) (Makki et al., 2015; Zafari et al., 2019) or sometimes, Time of Arrival (ToA) (Liu et al., 2007; Zhu et al., 2014). This method employs the simple relationship between signal travel time and the distance

$$D = (t_2 - t_1) \cdot v \tag{2.1}$$

where $(t_2 - t_1)$ is the flight time of the signal, that is the time delta between departure and arrival, and v is the signal propagation speed. With sufficient measurements the position can be determined as the intersection of circles, for example by using the Least Squares Method (LSM)

$$F(x) = \sum_{i=1}^{n} \alpha_i^2 f_i^2(x)$$

where α_i is a weight parameter and, assuming (x_i, y_i) is the position of the RN, (x, y) is the unknown position, furthermore

$$f_i(x) = D_i - \sqrt{(x_i - x)^2 + (y_i - y)^2}$$

The ToF method has strict time synchronisation requirements between a RN and the mobile target. For example in systems where the propagation speed in the speed of light, 3 nanoseconds equates to about 1 meter.

A special case of ToF is called Return Time of Flight (RToF) (Liu et al., 2007; Zafari et al., 2019) or Round Trip Time (RTT). This method uses the "there and back again" principle. Assuming a signal was sent at t_1 by node 1, the signal is then received by node 2, which immediately sends an acknowledgement. This is then received by node 1 at t_2. Then equation 2.1 becomes

$$D_{RTT} = (t_2 - t_1) \cdot \frac{v}{2}$$

this reduces the need for time synchronisation between the nodes, since all time-stamping done by node 1.

Time Difference of Arrival The Time Difference of Arrial (TDoA) is a localization method that measures the difference of signal propagation time to/from several nodes, either by having the tracked object sending a signal to the RNs, or by having the RNs transmit simultaneously to the tracked object (Liu et al., 2007; Zhu et al., 2014; Makki et al., 2015; Zafari et al., 2019). The first method is the most common, as the second needs to overcome potential issues with signal collision.

Knowing these time differences, we know that for each measurement, the tracked object must lie somewhere on the hyperbola of constant range differences, expressed as

$$R_{i,j} = \sqrt{(x_i - x)^2 + (y_i - y)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2}$$

Again (x_i, y_i) and (x_j, y_j) is the known positions of the RNs. This can then be solved by linear regression, or by linearizing using a Taylor series for an iterative solution.

This method still requires strict time synchronisation between the RNs, however, it does not require any synchronisation with the tracked object.

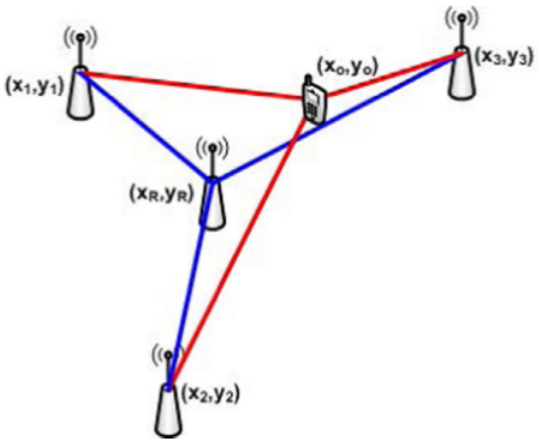


Figure 2.1: DTDoA localization (Makki et al., 2015)

Differential Time Difference of Arrival Makki et al. (2015) also mentions Differential Time Difference of Arrival (DTDoA). This method is vaguely similar to TDoA, however, the important distinction is that DTDoA only uses 1 RN, as illustrated in figure 2.1. The RN and receivers are all fixed and their position is known. The procedure starts with the RN transmitting a packet and the receivers noting the ToA. This first step also allows for time synchronisation between the APs. Then the tracked object transmits a packed and again the ToA is recorded by the receivers. Essentially, the system has now done two TDoA estimations, one for the RN and one for the tracked object. We can now use this information to solve an equation on the form

$$\Delta R_{i,j} = \left(\sqrt{(x_j - x_R)^2 + (y_j - y_R)^2} - \sqrt{(x_i - x_R)^2 + (y_i - y_R)^2}\right) - \left(\sqrt{(x_j - x)^2 + (y_j - y)^2} - \sqrt{(x_i - x)^2 + (y_i - y)^2}\right)$$

The main advantage with this method over basic TDoA, is that it does away with the need for additional time synchronisation between the APs. However, as most time based methods, it is prone to errors due to clock drift, especially if the delay between the RN and the tracked object is large.

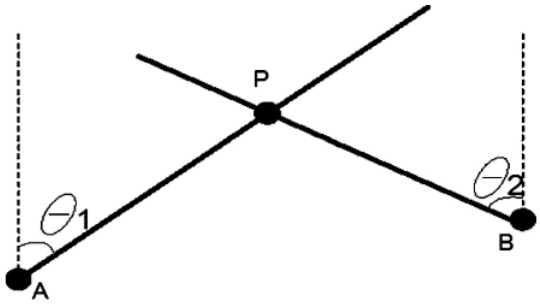


Figure 2.2: AoA based localization (Liu et al., 2007)

Angle of Arrival Angle of Arrival (AoA) is a method that estimates location based on the bearing towards known points (Liu et al., 2007; Zhu et al., 2014; Zafari et al., 2019) as illustrated in figure 2.2. The main advantages of this method is that only 2 RNs are needed for a 2D estimation and, that it does not require any time synchronisation. However, AoA based systems normally require specialized and often large hardware. Furthermore, systems are very sensitive to errors, as any small error in angle estimation will result in a large localization error.

Received Signal Strength Based Distance One of the simplest, and perhaps most widely used methods is distance estimation based on the Received Signal Strength Indicator (RSSI) (Liu et al., 2007; Zhu et al., 2014; Zafari et al., 2019). This is an estimation of the power of the received signal, measured in decibel-milliWatts (dBm) or milliWatts (mW), as reported by the hardware. There are various theoretical and empirical models for the relationship between RSSI and distance, but a common one is

$$RSSI = -10n \log_{10} d + A$$

where n is a path loss exponent and A is the RSSI at reference distance. This equation is then solved for distance d.

This method is both simple to implement and cost effective, as it does not require any special hardware to function. However, the accuracy is often poor, due to multipath problems, as the relationship between RSSI and distance is non-linear and is adversely affected by obstacles such as walls.

2.1.2 Scene Analysis

Another possible localization technique is called scene analysis or fingerprinting (Liu et al., 2007; Zafari et al., 2019). Normally, a number of fingerprint values, eg. RSSI or magnetic field information, is collected in an offline phase. This creates a map of fingerprints that online values, as observed by the tracked object, can be compared with. The offline phase of this method is often considered its main downside, as it is time consuming and therefore expensive. Additionally, it needs to be redone in every new location, as well as after any changes to the environment. Therefore, Jang and Kim (2019) presents a survey of systems which do not require this step by using methods such as inter/extrapolation or crowdsourcing. There is also a few different algorithms used to compare the offline and online values.

Probabilistic Method It is possible to consider positioning as a classification problem, based on the likelihood that a tracked object is in some location (x, y) given the observed values (Liu et al., 2007; Zafari et al., 2019). Consider a set of n location candidates

$L = \{L_1, L_1, \ldots, L_n\}$, then for any observed vector of values O, we select L_i as the location if

$$P(L_i|O) > P(L_j|O) \text{ for } i, j = 1, 2, \ldots, n \text{ and } i \neq j$$

or, assuming that $P(L_i) = P(L_j)$, we can use Bayes theorem to instead get the likelihood of vector O given location L_i, thus selecting L_i if

$$P(O|L_i) > P(O|L_j) \text{ for } i, j = 1, 2, \ldots, n \text{ and } i \neq j$$

The downside of this method is that it can only locate to discrete points, that is the points that was measured during the offline survey. However, it is possible to use a weighted average of positions instead, ie.

$$\hat{L} = \sum_{i=1}^{n} \Big[P(L_i|O)(L_i) \Big]$$

k-Nearest Neighbours In stead of calculating probabilities, k-Nearest Neighbours (kNN) finds the position based on the average position of the k nearest candidates in signal space (Liu et al., 2007; Caso et al., 2018; Zafari et al., 2019). The two important considerations with such a method is the selection of parameter k and the selection of similarity metric.

The most common similarity metrics are Root Mean Square Error (RMSE) or the Minkowski distance (D^p). Assume a set of fingerprints $F = \{F_1, F_2, \ldots, F_n\}$ and an observed vector O, then the similarity metrics between O and F_i is

$$RMSE = \sqrt{\sum_{j=1}^{n} \frac{(o_j - f_j^i)^2}{n}}$$

$$D^p = \Big(\sum_{j=1}^{n} |o_j - f_j^i|^p \Big)^{\frac{1}{p}}$$

where o_j and f_j^i is the j-th element of O and F_i respectively. Typically, $p = 1$, ie. Manhattan distance, or $p = 2$, ie. Euclidean distance.

Artificial Neural Networks Using techniques from machine learning, an Artificial Neural Network (ANN) can be trained to predict the location based on O (Liu et al., 2007; Zafari et al., 2019). The basic concept of a neural network is to find a set of weights W that parameterizes a function H_W such that the error of

$$\hat{L} = H_W(O)$$

is minimized, through training on a set of pre-classified data (Russell and Norvig, 2016). In the case of localization, this is the fingerprints from the offline survey and their corresponding positions.

The most common architecture of a localizing ANN is a multi-layer perceptron with one hidden layer, according to both Liu et al. (2007) and Zafari et al. (2019). However, this is not the only possible architecture, Shao et al. (2018) uses a Convolutional Neural Network (CNN) for a deep learning approach to fingerprinting.

Support Vector Machine A Support Vector Machine is another technique from the field of machine learning that can be used for localization purposes (Liu et al., 2007; Zafari et al., 2019). They have been used for classification and regression purposes in a wide variety of fields.

Smallest M-Vertex Polygon Smallest M-Vertex Polygon is a technique that searches for candidate locations by considering each fingerprint source, such as a WiFi or Bluetooth Access Point, individually (Liu et al., 2007). Each AP contributes at least one candidate, and then polygons with m vertices are created by selecting one candidate from all m sources. The average position of the *smallest* polygon is the estimated position.

2.2 Technologies & Systems

The previously mentioned techniques are not necessarily constrained to only one technology, in fact several different technologies have been investigated for use in indoor positioning, each with their own strengths and weaknesses. A summary of all the systems that are mentioned can be found in table 2.1.

Various sources divide systems into two main architectures (Liu et al., 2007; Yanying et al., 2009; Zhu et al., 2014; Zafari et al., 2019). The first is Device Based Localization (DBL). This architecture is characterized by the use of several transmitters in known points, with the receiver having unknown position. Location estimation is then performed on the device it self. This allows the mobile unit to ”own” its location resulting in better privacy. However, a potential downside is the potential reduced computational power of the receiver devices.

The other architecture is Monitor Based Localization (MBL). This can be considered to be the reverse architecture of DBL, in that is uses receivers in known points, while the mobile unit acts as a transmitter. The data is then sent to some central localization server that computes the final location estimation. This setup is more common in scenarios where the goal is to track inventory and/or personnel in some confined space. A common example is tracking expensive equipment in a hospital. In this case the tracked device no longer owns its own location, reducing potential privacy compared to DBL.

Liu et al. (2007) suggest that two additional architectures might exists. First is Indirect Monitor Based Localization (I-MBL). This uses a similar setup as *DBL*, but the location is then sent from the tracked device to a central server for storage, so that objects can

be tracked/monitored indirectly. This might alleviate some of the privacy concerns of standard MBL, if the tracked object is in control of communication with the remote server.

The other is Indirect Device Based Localization (I-DBL). This uses a similar setup as *MBL*, however, the location estimate is then sent to the tracked device, and not simply stored on the server, allowing the device to locate itself, just indirectly.

WiFi Also known as WLAN or IEEE 802.11, WiFi is an ubiquitous standard for high bandwidth communication between devices that operates on 2.4GHz and 5GHz frequencies of the Industrial, Scientific, and Medical (ISM) band (Liu et al., 2007; Zhu et al., 2014; Makki et al., 2015; Zafari et al., 2019). It is considered one of the prime technologies for indoor localization, since it is be possible to reuse hardware that is found in most buildings and devices today. However, it does have some problems with interference in the ISM band, that might affect accuracy.

One of the first systems to use WiFi for localization, was RADAR, developed by a Microsoft research group (Bahl and Padmanabhan, 2000). This system used the reported RSSI in two distinct ways to locate the device. The first is offline fingerprinting with a kNN algorithm. The other is modelling signal propagation using a Wall Attenuation Factor and a Floor Attenuation Factor in order to model the impact walls and floors have on RSSI. RADAR has a reported accuracy of 2-3 meters.

Horus (Youssef and Agrawala, 2008) is a fingerprinting based system that uses the probabilistic method previously described. It produces position estimations that are within 1.8 meters in 90% of cases, and a median accuracy as good as 39cm was achieved for one of the test beds. However, the use of fingerprinting does make it sensitive to changes in the environment.

Chronos (Vasisht et al., 2016) uses a single AP with multiple antennas to estimated the device location, requiring both devices to be Multiple-Input Multiple-Output devices, commonly known as MIMO. ToF is estimated to each of the AP antennas, and then an error minimization process is used to estimate the location of the device based on the geometric constraints of antenna configuration on the AP. Zafari et al. (2019) comments on the fact that, while the system achieves a median accuracy of 65cm, it is not scalable, and that it appears to have high power requirements.

Discussions about ToF based systems are left out of this section on purpose, as they will be presented and discussed in more detail in section 2.4.

Bluetooth Another Radio Frequency (RF) based and ubiquitous technology is Bluetooth, also known as IEEE 802.15.1. Like WiFi, it operates in the 2.4GHz range of the ISM band and is available on most phones and laptops (Liu et al., 2007; Zhu et al., 2014; Zafari et al., 2019). The latest iteration, called Bluetooth 5 or Bluetooth Low Energy

(BLE), can provide a data exchange rate of 24 Mbps and a range up to 100 meters. Furthermore, Angle of Arrival was added to the standard, possibly allowing sub meter accuracy in certain settings (Cominelli et al., 2019). But to date, most Bluetooth based systems use RSSI for location estimation.

iBeacons are a type of beacons based on Bluetooth that was developed by Apple Inc. specifically for proximity detection (Zafari et al., 2019). While using a protocol designed for proximity detection is certainly a good candidate for localization, it is a known limitation that, while the sampling rate is 50ms, only average RSSI is reported every second. This is a privacy decision made in the design of the protocol.

Topaz (as cited in Liu et al., 2007; Yanying et al., 2009) is an IPS based on Bluetooth. It has 3 main components: a positioning server, wireless APs, and Bluetooth tags. The Bluetooth enabled APs record the RSSI of nearby tags and forwards this to the positioning server for calculation. It provides room level accuracy; 2 meters with 95% reliability. However, a positioning delay of 15-30 seconds is incurred because of the system architecture.

Zafari (2016) developed a system where RSSI values of different iBeacons is recorded on the user device. This data is then forwarded to a server running 3 different algorithms; Particle Filter, Kalman Filter-Particle Filter, and Particle Filter-Extended Kalman Filter, used to improve the accuracy. The three approaches have a reported accuracy of 1.441m, 1.03m, and 0.95m respectively. The system is energy efficient but does suffer from a significant delay according to Zafari et al. (2019).

Zigbee Developed by the Zigbee Alliance and built upon the IEEE 802.15.4 standard, Zigbee is a low data rate, low cost, and energy efficient protocol for PNs, such as smart home devices (Zafari et al., 2019). This makes it a possible candidate for localization in Wireless Sensor Networks, however, it has not (yet) seen the same widespread use as WiFi and Bluetooth in user devices, making it less favorable compared to these. Uradzinski et al. (2017) uses Zigbee along with a fingerprint database to achieve sub-meter accuracy.

Radio Frequency Identification A technology used in many localization systems is RFID (Liu et al., 2007; Yanying et al., 2009; Zafari et al., 2019). It is a technology for transmitting and storing small data using electromagnetic transmission. Systems consists of two parts, the RFID tag which stores and emits the data (often an ID), and a reader that can read or write to the tags. It can be for example be used as a replacement for barcodes.

Two basic variants exists, defined by the types of tags that are used. First is *active* RFID that operate in the Ultra High Frequency, known as UHF, or microwave range. In these systems, the tags have their own power source and periodically transmit their ID. They have reasonable range and tags are easy to embed in devices.

The second type is *passive* RFID, where tags do not have their own power source, and as such are smaller, lighter, and cheaper than their active counterparts. Communications is performed when the reader inducts current into the passive circuit, initiating transmission. This system has very limited range, 1-2 meters, making them mostly unsuitable for indoor localization purposes.

WhereNet (as cited in Yanying et al., 2009) is a system developed by Zebra Technology company. It uses a TDoA algorithm, where antennas in fixed, known locations receive signals form active tags and forward this information to a central server for calculations. The system has a reported accuracy of 2-3 meters.

LANDMARC (Ni et al., 2004) uses additional tags as reference points, to increase accuracy without the need to increase the number of readers. Then the kNN algorithm is used to estimate location on a central server. LANDMARC is energy efficient, and has decent range. However, the system has high tracking latency. The median accuracy is 1 meter.

Shirehjini et al. (2012) propose a system that utilizes a carpet of passive RFID tags. Readers on the mobile unit can read the information from nearby tags and uses it to compute location. The proposed system is very accurate, with average errors of 6.5cm. However, using low range tags necessitates the use of a large number of them.

Ultra Wide Band UWB uses ultra short pulses ($>$ 1 nanosecond) and large bandwidth ($>$ 500MHz) allowing it to achieve low power consumption and, more importantly, be very robust to the multipath problem, as the short pulses make it easier to identify the first arrival of the signal while filtering out subsequent ones, caused by signal reflection (Liu et al., 2007; Yanying et al., 2009; Zafari et al., 2019). This short pulse duration also allows for accurate ToF estimations. Another significant advantage is that UWB does not suffer from interference from RF signals, such as WiFi or Bluetooth, due to differences in signal types, as well as the radio spectrum used. Furthermore, it can penetrate a wide variety of materials, including walls, but does suffer from interference from liquids or metallic materials. Furthermore, as a result of the slow progress on a UWB standard, it sees only limited use in consumer products.

Ubisense (as cited in Liu et al., 2007; Yanying et al., 2009; Zafari et al., 2019) is a commercial positioning system from the Ubisense Company that was founded by researchers from AT&T Cambridge. It takes advantage of both the TDoA and AoA to estimate location. The system consists of three parts: Sensors, Tags, and a proprietary software platform. While it achieves an accuracy as high as 15cm, high cost is a limiting factor for this system.

Ultrasound Best known as the mechanism used by bats to navigate in caves, ultrasound have been investigated for positioning purposes (Liu et al., 2007; Zafari et al., 2019). These

systems are generally based on ToF as the lower signal speed, ie. the speed of sound, results in lower requirements for time fidelity. As seen in figure 2.3, ultrasound does not penetrate walls like RF signals do, meaning that systems are very applicable to room level tracking. Additionally, systems have been shown to have centimeter accuracy. Downsides to such systems include being susceptible to reflected signals and noise caused by metallic objects. Furthermore, the speed of sound will change depending on temperature and humidity, requiring additional sensors to compensate.

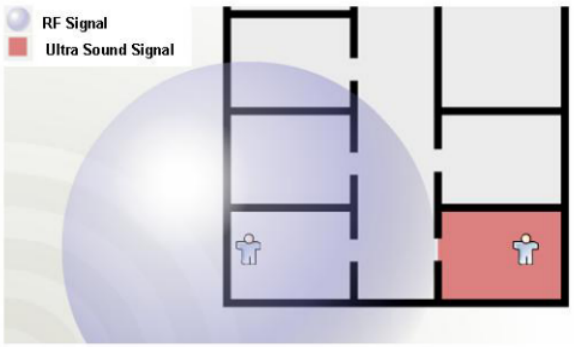


Figure 2.3: Comparison between RF and Ultrasound propagation (Yanying et al., 2009)

Cricket (Priyantha et al., 2000) is an IPS that uses ultrasound and ToF to locate a target. Emitters are mounted in known locations on walls and ceilings, while each tracked object is fitted with a receiver, which also performs the triangulation. Localization is performed by an emitter that concurrently transmits an RF signal and a ultrasound pulse. When a receiver detects the RF signal, it then begins to listen for the ultrasound pulse that arrives a little later. The difference in arrival time between these to signals is then used to estimate the ToF. The reported accuracy of the system is 10cm.

The Bat system (Harle and Hopper, 2005) is developed jointly by AT&T Cambridge and the University of Cambridge. It uses a system of active tags called bats that is worn by the users. Upon receiving instructions to do so, a bat transmits an ultrasonic signal that is then received by a matrix of receivers in the ceiling. Each receiver has a known location, such that the location of the bat can be calculated. The positional accuracy has been determined to be within 3cm in 95% of cases.

Audible Sound In a similar fashion to ultrasound, some positioning systems leverage sound signals in the audible range (Yanying et al., 2009; Zafari et al., 2019). Systems can either rely on tracked objects emitting an acoustic signal that can be detected by microphones in the environment, or microphones on the tracked object can detect signals emitted from RNs. Either way, both microphones and loudspeakers are ubiquitous on smartphones and laptops, allowing for the reuse of these components. While Yanying et al. (2009) argues that using signals that are perceivable to the human ear increases privacy by making it explicit when an object is tracked, Zafari et al. (2019) argues that

signals should be of low enough power to be imperceptible so that they do not cause noise pollution.

Beep (Mandal et al., 2005) uses sound produced by a mobile device and the ToF technique to locate objects. Acoustic sensors in fixed positions detect and forward the signal to a central server which computes the location over WiFi. WiFi is also used in order to time synchronize the sensors and the mobile device. The reported accuracy is 0.6 meters in 97% of cases.

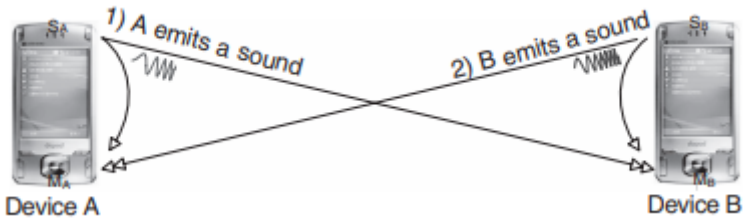


Figure 2.4: The BeepBeep system (Peng et al., 2012)

Not to be confused with the previous system, BeepBeep (Peng et al., 2012) uses an acoustic signal and commodity hardware to enable two way ranging. This enables ranging to be performed between any two devices that have a microphone and a loudspeaker. As shown in figure 2.4, the procedure has two steps. First device A will emit a sound to be recorded by both A and B. Then device B will emit a different sound to be recorded by both B and A. Each device now holds a recording containing both signals, and can use this to compute the ToF to the other device. Reported accuracy is 1cm for ranging and 3cm for positioning.

GuoGuo (Liu et al., 2016) is an example of a sound based system that uses an acoustic signal imperceptible to the human ear. To this end, it requires proprietary acoustic transmitters, while the tracked device uses advanced signal processing in order to detect the signals. The reported accuracy of the system is 6-15cm.

Infrared IR is a form of electromagnetic radiation that has been suggested for several positioning systems (Yanying et al., 2009). On one hand, these systems are considered to be very accurate, with tags that are small, light-weight and cheap. On the other, there are some important disadvantages, including a LOS requirement and being susceptible to interference from sunlight, as well as florescent lighting. And although tags are light and low cost, the camera arrays used to detect them can be expensive, especially compared to coverage area. Finally, IR was considered a good candidate because of its widespread use in various devices. These days however, the use of IR is mainly limited to remote controls.

Active Badge (Want et al., 1992) is another system developed by AT&T Cambridge. Badges carried by users or attached to tracked objects transmit a unique IR signal every

15 seconds, that is then detected by at least one sensor in the space. The accuracy of the system is room level only.

Firefly (as cited in Yanying et al., 2009) is a motion tracking system, developed by Cybernet System Corporation, that can be used for location purposes. It consists of tags, a controller, and a camera array, and uses the IR light emitted by several tags that are attached to the object. This allows the system to track and animate the motion of that object. The tags are lightweight and small, but must be attached to the controller using wires. Furthermore, the area covered by the camera array is small, about 7 meters. On the other hand, the accuracy is reported to be about 3mm.

Visible Light Visible Light Communication is an emerging technology for high speed data transfer (Zafari et al., 2019). It uses Light Emitting Diodes (LEDs) to emit visible light in the 400-800THz range, which can be detected by sensors which can then estimate the position of, and direction towards the LEDs, with AoA being considered to be the most accurate method.

Other Besides the major technologies discussed above, there are also examples of systems that does not rely on either of them, or even some combination of them.

Shao et al. (2018) uses both WiFi and magnetic fingerprints to estimate location. They create images from the fingerprint information of the two mediums and combines them using a CNN. The reported accuracy is 1 meter in 95% of cases.

GROPING (Zhang et al., 2015) uses crowdsourcing to construct a geomagnetic map of a floor or a building that can then be used to localize users. However, the cited accuracy is 5 meters in 90% of cases, and the system uses 30 seconds to converge on this answer, so it cannot work in real time.

Easy Living (Brumitt et al., 2002) is a vision based position system developed by a Microsoft research group. It uses two cameras to track objects within an area. The benefit of this method is that there is no hardware requirement on the user side. However, the accuracy is only room level.

Lu et al. (2016) proposes a system that uses thermal imaging to estimate user location. Thermal imaging has the benefit, as compared to traditional imaging, that the system will function even in the absence of light, which might be useful in search and rescue scenarios. The authors only report the classification accuracy of the system, it classifies the correct location in about 97% of cases.

Name/Author	**Technology**	**Accuracy**	**Source**	**Part of Survey**
RADAR	WiFi	2-3 meters	Bahl and Padmanabhan (2000)	Liu et al. (2007) Yanying et al. (2009) Zafari et al. (2019)
Horus	WiFi	1.8 meters in 90% of cases	Youssef and Agrawala (2008)	Liu et al. (2007) Zafari et al. (2019)
Chronos	WiFi	65cm median	Vasisht et al. (2016)	Zafari et al. (2019)
Topaz	Bluetooth	2 meters in 95% of cases	Commercial	Liu et al. (2007) Yanying et al. (2009)
Zafari	Bluetooth/iBeacon	0.9-1.4 meters depending on filter	Zafari (2016)	Zafari et al. (2019)
Uradzinski et al.	Zigbee	sub-meter	Uradzinski et al. (2017)	None
WhereNet	RFID	2-3 meters	Commercial	Yanying et al. (2009)
LANDMARC	RFID	1 meter median	Ni et al. (2004)	Liu et al. (2007) Zafari et al. (2019)
Shirehjini et al.	RFID	6.5cm	Shirehjini et al. (2012)	Zafari et al. (2019)
Ubisense	UWB	As high as 15cm	Commercial	Liu et al. (2007) Yanying et al. (2009) Zafari et al. (2019)
Cricket	Ultrasound	10cm	Priyantha et al. (2000)	Yanying et al. (2009) Zafari et al. (2019)
Bat System/Active Bat	Ultrasound	3cm in 95% of cases	Harle and Hopper (2005)	Yanying et al. (2009) Zafari et al. (2019)
Beep	Audible Sound	0.6 meters in 97% of cases	Mandal et al. (2005)	Yanying et al. (2009) Zafari et al. (2019)
BeepBeep	Audible Sound	3cm	Peng et al. (2012)	Zafari et al. (2019)
GouGou	Audible Sound	6-15cm	Liu et al. (2016)	Zafari et al. (2019)
Active Badge	Infrared	Room level	Want et al. (1992)	Yanying et al. (2009)
Firefly	Infrared	3mm	Commercial	Yanying et al. (2009)
Shao et al.	Magnetic Field and WiFi	1 meter in 95% of cases	Shao et al. (2018)	None
GROPING	Magnetic Field	5 meters in 90% of cases	Zhang et al. (2015)	Zafari et al. (2019)
Easy Living	Visual	Room level	Brumitt et al. (2002)	Yanying et al. (2009)
Lu et al.	Thermal Image	Correct classification in 97% of cases	Lu et al. (2016)	Zafari et al. (2019)

Table 2.1: Summary of systems

2.3 Time Based Ranging in WiFi

As mentioned previously, WiFi is currently considered the prime technology for consumer facing indoor positioning, because of its wide spread use both in buildings and on user devices. While much of the work up until today has been focused around RSSI, such as the previously mentioned RADAR, considered a pioneering work in the field, the accuracy of such methods is known to be limited (Au, 2016). Therefore, work has also been done to investigate the feasibility of a time based system using WiFi. This work culminated in the Fine Timing Measurement protocol being added to in the IEEE 802.11-2016 amendment of the WiFi standard (IEEE, 2016). This section will first introduce some of the preliminary work on time based ranging in WiFi, then describe the official FTM standard, then a least squares algorithm, used to estimate coordinate position from the distance ranges, will be introduced. Finally, related work using FTM will be presented.

2.3.1 Preliminary Work

While there has been substantial work put into researching time based methods for WiFi, this section will only focus on a subset of them. Makki et al. (2015) presents a more varied survey of this work, as well as a good overview of the different iterations of the IEEE 802.11 standard.

An important first step in enabling time based ranging, is the work of Gunther and Hoene (2005), which proved that it was possible to use off-the-shelf WiFi hardware to estimate range, despite the low resolution of the timestamps provided at that time. The technique used the DATA and ACK (acknowledge) packets, as well as statistical methods in the application layer. The reported mean ranging error is 8 meters, and required 1000 packet transmissions for the statistical methods to be effective.

Hoene and Willmann (2008) built on the work of Gunther and Hoene, using the Request-to-Send (RTS) and Clear-to-Send (CTS) packets in addition to DATA and ACK. This reduces the number of packet sequences required, while also increasing accuracy to about 4 meters.

While these initial investigations into application level tracking showed that ranging was indeed feasible, they are clearly limited in their accuracy, as well as requiring a large number of packets. To address this problem, Izquierdo et al. (2006) presents a method that works on the lower MAC level by using the built in 44 MHz clock of the WiFi card to do time-stamping. This both increases the time resolution, as well as eliminating any delay that might occur between receiving a result and that result being communicated to the application layer. Furthermore, they only need to use the RTS and CTS packets. Using 300 packet sequences and a least squares method, similar to the one presented in 2.3.3, they report an accuracy of 2 to 2.3 meters in 90% of cases.

Some of the authors expand on this work to use a Discrete Kalman Filter in dy-

namic tracking environments (Ciurana et al., 2006). Using simulations, they find that the tracking error of their Kalman filter is 1.4m in 90% of cases.

Ciurana et al. (2009) further explored using the Central Processing Unit (CPU) clock instead of the WiFi card clock, by modifying the driver software to use the CPU Time Stamp Counter (TSC). In general, the CPU is configured at a much higher frequency than the WiFi card, normally $>$ 1GHz, resulting in $<$ 1ns time resolution. This method uses the DATA-ACK packet sequences and is reported to have an accuracy better than 2 meters in LOS environments.

Schauer et al. (2013) similarly used the CPU clock, however they preferred to use the NULL-ACK sequences for their ranging. Using only 30 RTT values, they achieved a mean ranging error of 1.3 meters using an HP laptop in a optimal, ie. LOS, environment. However, the same procedure on a Samsung laptop had a mean error of 27 meters. The procedure was also tested in an office environment, where the respective systems produces a mean error of 5.5 and 275 meters respectively. The results clearly show that the hardware that is used has a significant impact on the accuracy, all things being equal.

Casacuberta and Ramirez (2012) compare the previously described method of using the CPU clock, with a timestamp from the Timing Synchronisation Function (TSF) that is used to coarsely synchronise WiFi equipment on the same network. They find that the CPU TSC method is better, with a mean error 1.5 meters compared to 2.8 meters.

Ciurana et al. (2011) looked at the benefits to then then new IEEE 802.11v amendment. First, the requirement for nodes to be associated or authenticated before exchanging frames is removed, thus allowing any device to communicate with any AP in such a way as to allow for ranging without delay. Second, the AP processing time, ie. the turnaround time, is added to responses with an expected resolution of up to a tenth of a millisecond. It is vital to know this delay, since the speed of light is approximately 0.3 meters per *nanosecond*, so even a minimal delay will cause significant errors in the range estimation. Previously, this delay had to be determined empirically by measuring the delay when immediately next to the AP. Finally, 802.11v was expected to include timestamps of the transmission and reception of frames. This became the Timing Measurement protocol, the predecessor to FTM.

A slightly different, more low level method using a time-continuous Barker code is introduced by König et al. (2010). This method uses signal processing rather than time-stamping, estimating the ToF as the delay of the peak correlation of the Barker code. The advantage of this is that it provides "sub-sample" accuracy, that is accuracy beyond the sampling rate of the WiFi card itself. The authors report an accuracy of 1.17 meters in a LOS environment.

2.3.2 WiFi Round-Trip-Time

To simplify and standardize time based localization using WiFi, the FTM protocol of the 802.11mc amendment was added to the WiFi standard in the 802.11-2016 revision (IEEE, 2016). This protocol provides a standardized way to estimate distances between nodes in a WiFi network by measuring the RTT of a signal.

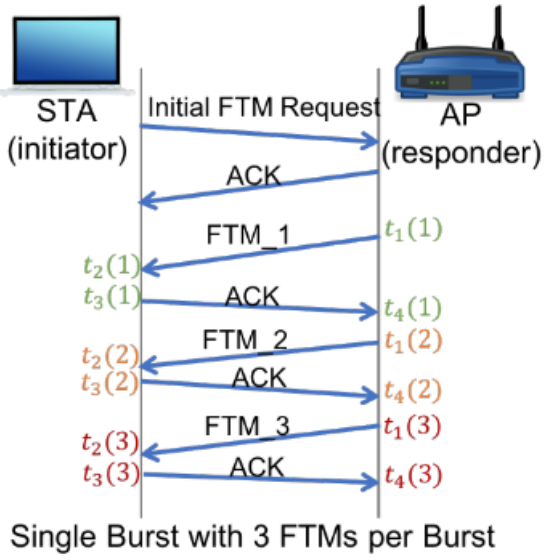


Figure 2.5: Basic overview of FTM protocol (Ibrahim et al., 2018)

Figure 2.5 illustrates the basic principle of the FTM protocol. An initiator, eg. a smartphone, starts by sending a FTM request. This is acknowledged by the responder, eg. a WiFi Access Point, indicating that the responder is ready to send FTM packets. One request initiates one burst of n measurements.

The responder then sends a FTM packed with the exact time of departure t_1 indicated. Then, the initiator records the exact time of arrival t_2 and immediately responds with a packet containing these timestamps, as well as the time of departure for this packet t_3. Finally, the responder records the time of arrival t_4. Assuming n such measurements the RTT can be calculated as

$$RTT = \frac{1}{n}\Big(\sum_{i=1}^{n} t_4(i) - \sum_{i=1}^{n} t_1(i)\Big) - \frac{1}{n}\Big(\sum_{i=1}^{n} t_3(i) - \sum_{i=1}^{n} t_2(i)\Big) \tag{2.2}$$

As mentioned before, it is important to know the processing delay to perform accurate ranging. The FTM protocol handles this by recording the t_2 and t_3 timestamps in addition to start and end times. As can be seen in equation 2.2, the processing delay can then be subtracted, leaving only the actual flight time of the signal.

Now that we have the total flight time of the signal the range between the two nodes can be calculated as

$$D^{RTT} = \frac{RTT}{2c}$$

where c is the speed of light.

2.3.3 Least Squares Method

Having obtained ranges to at least 3 AP's, that is 3 known locations, we can use the process of LSM in order to estimate the location of the receiver node (Dargie and Poellabauer, 2010). This is done by estimating the intersection of circles. Given the circle equation

$$x^2 + y^2 = r^2$$

we can construct the following set of equations

$$\begin{bmatrix} (x_1 - x)^2 + (y_1 - y)^2 \\ (x_2 - x)^2 + (y_2 - y)^2 \\ \vdots \\ (x_n - x)^2 + (y_n - y)^2 \end{bmatrix} = \begin{bmatrix} r_1^2 \\ r_2^2 \\ \vdots \\ r_n^2 \end{bmatrix}$$

where (x_i, y_i) is the known position of node i, and r_i is the estimated distance to the node, ie. D_i^{RTT}.

This system can be expressed as a linear least squares problem on the form

$$A\mathbf{x} = b + v \tag{2.3}$$

where v is the vector of residuals, and

$$A = 2 \cdot \begin{bmatrix} (x_n - x_1) & (y_n - y_1) \\ (x_n - x_2) & (y_n - y_2) \\ \vdots & \vdots \\ (x_n - x_{n-1}) & (y_n - y_{n-1}) \end{bmatrix}$$

and the left hand side

$$b = \begin{bmatrix} r_1^2 - r_n^2 - (x_1^2 - x_n^2) - (y_1^2 - y_n^2) \\ r_2^2 - r_n^2 - (x_2^2 - x_n^2) - (y_2^2 - y_n^2) \\ \vdots \\ r_{n-1}^2 - r_n^2 - (x_{n-1}^2 - x_n^2) - (y_{n-1}^2 - y_n^2) \end{bmatrix}$$

and the solution to this least squares system is then

$$\mathbf{x} = (A^T W A)^{-1} A^T W b$$

where W is the weight matrix.

The weights can either be the identity matrix, equivalent to unweighted LSM, or it can be weights based on the measurements. Dargie and Poellabauer (2010) suggest the following formula

$$w_i = 1/\sqrt{\sigma^2_{distance_i} + \sigma^2_{position_i}}$$

where $\sigma_{distance_i}$ is the reported standard deviation of the range to the ith AP, and $\sigma_{position_i}$ is the standard deviation of the known position of the AP. Another alternative from surveying is weighting on the measured distance, putting more emphasis on the results from RNs that are closer. (Ghilani, 2010)

$$w_i = 1/r_i$$

This method is very useful for calculating an initial position for the unknown point, however, this is not very accurate. In traditional surveying it is common to use a non-linear least squares based on Newtons method in order to solve the problem iteratively (Ghilani, 2010). This method takes an initial "guess" for the location and iteratively calculates corrections to the estimate, until the corrections are below some threshold. While the individual equations change, the basic matrix representation has the exact same form as equation 2.3

$$J\Delta\mathbf{x} = k + v \tag{2.4}$$

where J is the Jacobian matrix of partial derivatives, $\Delta\mathbf{x}$ is the vector of unknowns, ie. the corrections, k is a vector of constants, and v is the vector of residuals. Specifically for our case we have that

$$J = \begin{bmatrix} \frac{\hat{x} - x_1}{\hat{r}_1} & \frac{\hat{y} - y_1}{\hat{r}_1} \\ \frac{\hat{x} - x_2}{\hat{r}_2} & \frac{\hat{y} - y_2}{\hat{r}_2} \\ \vdots & \vdots \\ \frac{\hat{x} - x_n}{\hat{r}_n} & \frac{\hat{y} - y_n}{\hat{r}_n} \end{bmatrix}$$

and

$$k = \begin{bmatrix} r_1 - \hat{r}_1 \\ r_2 - \hat{r}_2 \\ \vdots \\ r_n - \hat{r}_n \end{bmatrix}$$

where $(\hat{x}, \hat{y})$ is the current best estimate for the unknown position, (x_i, y_i) is the known position of the ith AP, r_i is the range observed between the receiver and the AP, while $\hat{r}_i$ is the same range calculated using the known coordinates of the ith AP and the estimated location of the unknown point.

The solution to this system is then

$$\Delta\mathbf{x}_j = (J^T W J)^{-1} J^T W k$$

and

$$\hat{\mathbf{x}}_{j+1} = \hat{\mathbf{x}}_j + \Delta\mathbf{x}_j$$

where $\hat{\mathbf{x}}_j$ is the jth estimation of the position.

Using LSM we can also calculate the precision of our estimates, by calculating the standard deviation (Ghilani, 2010). Remember that the two equations 2.3 and 2.4 have the exact same form, so the following is valid for both methods, but will use the variable naming from 2.3.

Equation 2.3 can alternatively be written as

$$v = A\mathbf{x} - b$$

giving us the residuals v of the current estimation. From this we can estimate the reference standard deviation S_0 as

$$S_0 = \sqrt{\frac{v^T W v}{n - e}}$$

where $n - e$ is the number of observations minus the number of unknowns, ie. the degrees of freedom in the system. We can then calculate the variance-covariance matrix as

$$S_{xx}^2 = S_0^2 \cdot (A^T W A)^{-1} = \begin{bmatrix} S_{x_1}^2 & S_{x_1 x_2} & \dots & S_{x_1 x_n} \\ S_{x_2 x_1} & S_{x_2}^2 & \dots & S_{x_2 x_n} \\ \vdots & \vdots & \ddots & \vdots \\ S_{x_n x_1} & S_{x_n x_2} & \dots & S_{x_n}^2 \end{bmatrix}$$

where $S_{x_i x_j}$ is the covariance between the ith and jth unknowns, and $S_{x_i}^2$ is the variance of the ith unknown.

2.4 Related Work

Naturally, the FTM protocol have been considered in various other works, both to verify its accuracy, as this study will, or by fusing the results of FTM with other positioning techniques.

Ibrahim et al. (2018) also attempts to verify the meter-level accuracy claim. They use Intel Dual Band Wireless-AC 8260 and 8265 wireless chips in both client and AP mode, as well as ASUS Wireless-AC1300 RT-ACRH13 APs. They conducted experiments both in outdoor environments, as well as indoor. For our purposes, the indoor results are the most relevant.

For distance observations, they observe a 2.6 meter median error, and a 6.5 90th percentile error. They also propose a temporal filtering technique that improves the results to 2.5 and 4.8 meters. They also check localization accuracy by using an iterative non linear least squares algorithm similar to that previously presented in this chapter. In this case, they report the accuracies for each type of AP separately. For the Intel cards, the median accuracy is 5.2 meters and 90th percentile of 11.6. After applying the filtering, these become 4.2 and 8.2 respectively. Using the ASUS APs, the reported accuracies are

	Ranging	Ranging Filtered	ASUS Location	ASUS Filtered	Intel Location	Intel Filtered
Mean	2.6	2.5	3.8	3.5	5.2	4.2
90th	6.5	4.8	6.2	4.7	11.6	8.2

Table 2.2: Summary of Ibrahim et al. errors

3.8 median and 6.2 90th percentile. Again the filtering improves these results to 3.5 and 4.7 respectively.

Overall they conclude that, while meter level accuracy is possible in some specific conditions, the accuracy in rich multipath environments does not seem higher than that demonstrated by other systems. However, they note that FTM can deliver this accuracy with relatively few access points and minimal site survey, and that accuracy should improve with a denser deployment of APs.

Šeleng (2019) compares range estimations of FTM with range estimations based on Bluetooth and WiFi RSSI. As expected FTM outperforms these methods when tested on a Google Pixel 2. The reported average error of FTM is 1.7 meters in one environment and 1.2 meters in another. Location accuracy is not tested.

Xu et al. (2019) uses data from smartphone sensors to perform Pedestrian Dead Reckoning (PDR), and then use a particle filter to fuse these observations with FTM. The PDR also helps detect outliers in the FTM data. The authors analyse both FTM ranging errors, as well as comparing the localization errors of FTM and a linearized least squares, PDR alone, and the proposed particle filter fusion method. In the study, a Google Pixel 3 smartphone is used, but no indication to the type of AP is given.

Using FTM ranging alone, the authors report an error less than 1.35 meters in 80% of cases. However, they also observe errors as high as 4 meters in some Non-Line of Sight (NLOS) scenarios. While for localization, the authors report a mean error of 1.65 meters for PDR, 2.21 meters for LSM, and 0.89 for the fusion method.

The authors conclude that using PDR alongside FTM is a viable option for indoor localization, and provides meter level accuracy using 2000 particles in the filter, with a positioning delay of about 0.5 seconds.

Yu et al. (2019) present a similar approach, but using a Unscented Kalman Filter instead. They report all errors to be within 2 meters.

Finally, Guo et al. (2019) proposes to combine FTM with classical RSSI scene analysis methods, fusing them with a scalar Kalman filter. Their experiments used a Intel Dual Band Wireless-AC 8260 WiFi card, and used both a Google Pixel and Google Pixel 3 smartphone.

While they test their algorithm in both outdoor and indoor environments, only the results in indoor environment is interesting to this thesis. Using the raw data from FTM,

they report a mean ranging error of 0.896 meters, as well as a 95th percentile error of 1.838 meters. Their proposed fusion method achieves 0.443 meters mean error, and 95th percentile of 1.161 meters.

The authors also report on the localization accuracy of their proposed algorithm. In this case the mean error using FTM only is 2.063, and less than 4.918 in 95% of cases. The same metrics for their proposed algorithm is 1.435 and 2.761 respectively.

Chapter 3

Experiment

In order to verify the accuracy of WiFi RTT, several test was conducted. At first, the possibility to use the existing WiFi network at NTNU was investigated. Unfortunately, the APs currently used in the network does not yet support FTM. Instead it was decided to use a Google Pixel 3a mobile phone and 3 Google WiFi routers/APs, in combination with the indoor network of benchmarks that exists in the building of the Department of Civil and Environmental Engineering at NTNU. This chapter will describe the setup and execution of these test.

3.1 Indoor Control Points

Figure 3.1: Example of benchmark on linoleum floor

In 2012, a network of control points was established in connection with previous experiments on indoor positioning at the department (Midtbø et al., 2012). These are points with known coordinates in a coordinate system specific to the building, and was therefore used as ground truth values during the tests. The points are marked in the linoleum floor

of the building with a physical imprint, as well as red enamel paint, as illustrated in figure 3.1

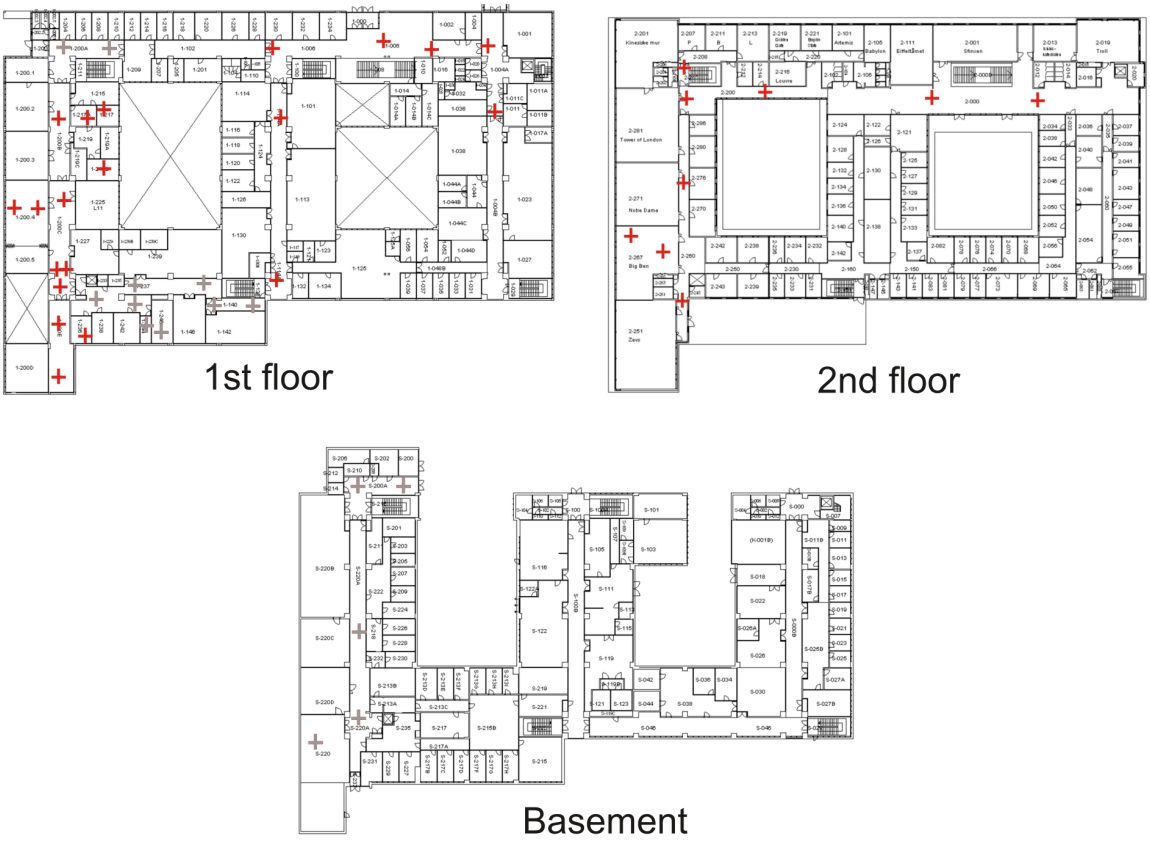


Figure 3.2: Locations of the control points within the building

During the Fall semester of 2019, renovations was started in some parts of the building, and were still ongoing as of the Spring semester of 2020, during which work on this thesis was performed. As a result of the renovations, a number of the control points was destroyed or otherwise made unavailable. Figure 3.2 shows the location of all control points within the building, including the aforementioned unavailable ones, which are indicated with a grey cross. Furthermore, a larger version which includes point name labels can be found in appendix A, along with the coordinates of each point.

3.2 Android Application

For the purposes of conducting the actual tests, an Android application was developed to run on the Pixel 3a hardware. This smartphone was selected specifically because it is one of few current models that is known to support the WiFi FTM protocol.

3.2.1 RTT API

Android provides an API[1] that allows you to do ranging based on WiFi FTM using compatible mobile phones. This section will give a brief introduction to the API, with code examples in the Kotlin programming language.

In order to use the API, a total of three permissions are required, as these inform and protects the user's privacy. On the Android platform, such permissions are declared in `AndroidManifest.xml` as shown

Listing 3.1: Define permissions

```
<uses-permission
    android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission
    android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission
    android:name="android.permission.ACCESS_FINE_LOCATION" />
```

Furthermore, the `ACCESS_FINE_LOCATION` is considered a *dangerous permission* by the system, and must therefore be explicitly requested at runtime, as shown

Listing 3.2: Check and request permission

```
private fun checkPermission(): Boolean
    if (
        ContextCompat.checkSelfPermissions(
            context,
            Manifest.permission.ACCESS_FINE_LOCATION
        ) != PackageManager.PERMISSON_GRANTED
    ) {
        // Permission not already granted
        ActivityCompat.requestPermissions(
            activity,
            arrayOf(Manifest.permission.ACCESS_FINE_LOCATION),
            requestCode)
        return false
    } else {
        // Permission was already granted
        return true
    }
```

This function will present the user with a dialog box, with options to grant or deny location permission. In order to receive the result of this prompt, applications should override the `onRequestPermissionResult` function, as shown

Listing 3.3: Capture permission result

```
override fun onRequestPermissionsResult(
    requestCode: Int,
    permissions: Array<out String>,
```

[1] https://developer.android.com/guide/topics/connectivity/wifi-rtt

```
    grantResults: IntArray
) {
    // Assuming ACCESS_FINE_LOCATION is the only dangerous
       permission
    if (
        grantResults.isNotEmpty() &&
        grantResults[0] == PackageManager.PERMISSION_GRANTED
    ) {
        // Access granted
    }
}
```

When the relevant permissions have been granted by the user, the application needs to perform a scan for nearby access points that support the FTM standard. This is done by constructing a callback receiver, registering for the relevant intent with the system, and then requesting a scan to be performed.

Listing 3.4: Scan for compliant APs

```
private fun doScan() {
    // Get WiFi system service
    val wifiManager = activity.getSystemService(Context.
       WIFI_SERVICE) as WifiManager

    // Set up result receiver
    val wifiScanReceiver = object : BroadcastReceiver() {

        override fun onReceive(context: Context, intent: Intent)
           {
            // We do no require subsequent results
            context.unregisterReceiver(this)

            // Check if scan was successful
            val success = intent.getBooleanExtra(
                WifiManager.EXTRA_RESULTS_UPDATED,
                false
            )

            if (success) {
                // Filter scan results and save
                networklist = wifiManager
                    .scanResults
                    .filter{ it.is80211mcResponder }
            } else {
                // Handle scan error
            }
        }

    }

    // Register for the SCAN_RESULTS_AVAILABLE_ACTION intent
    val filter = IntentFilter()
```

```
    filter.addAction(WifiManager.SCAN_RESULTS_AVAILABLE_ACTION)
    context.registerReceiver(wifiScanReceiver, intentfilter)

    // Initiate scan
    wifiManager.startScan()
}
```

Note the `ScanResult#is80211mcResponder` field, which indicates whether the AP of the corresponding `ScanResult` supports WiFi FTM. There is also a limit on how often any application can initiate a scan, approximately 4 times over 2 minutes. Scanning more often than this will result in a scan error.

Now that a list of the nearby compliant APs has been obtained, the application is ready to perform ranging. To do this it uses the `WifiRttManager` to initiate a single burst to each of the APs in `networklist`.

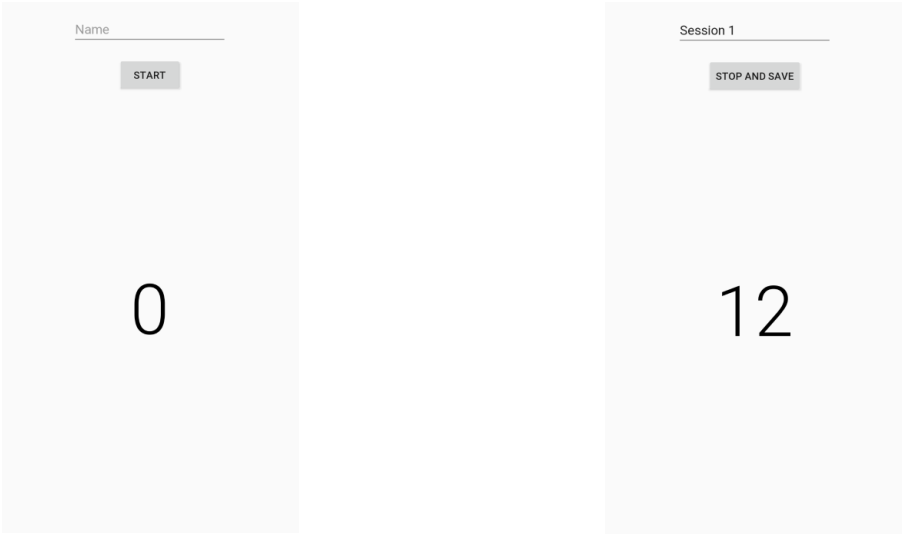
Listing 3.5: Perform ranging

```
private fun doRanging() {
    if (!context.packageManager.hasSystemFeature(PackageManager.
       FEATURE_WIFI_RTT)) {
        // RTT not available on phone
        return
    }

    // Get RTT system service
    val rttManager = context.getSystemService(Context.
       WIFI_RTT_RANGING_SERVICE) as WifiRttManager

    if (!rttManager.isAvailable) {
        // RTT disabled at the moment
        return
    }

    // Build a request with all APs
    val request = RangingRequest.Builder().run {
        networklist.forEach { addAccessPoint(it) }
        build()
    }

    // Create callback receiver
    val callback = object : RangingResultCallback() {

        override fun onRanginResults(res: List<RangingResult>) {
            // Handle ranging result
        }

        override fun onRangingFailure(res: Int) {
            // Handle ranging error
        }

    }
```

```
    // Perform SINGLE burst
    rttManager.startRanging(
        request,
        AsyncTask.THREAD_POOL_EXECUTOR,
        callback
    )
}
```

3.2.2 Implementation

The application[2] was developed using the built-in API that was presented in the previous section. The simple application allows a user to scan for AP's within range that support the protocol and then select any number of them to perform ranging.



(a) Ranging view before starting (b) Ranging view during measurement

Figure 3.3: Example of ranging in application

When the relevant APs have been selected, the user is presented with a screen that allows to set a name/description to the session, before selecting start. Ranging bursts are then performed at the rate of once pr second. When ranging is stopped, the results are saved in a predefined `.csv` file format in internal storage. The columns of the file are

`mac_address` The mac address/BSSID of the AP in question

`timestamp` Time when the ranging was performed in millis since startup

`attempted` Number of rangings attempted in burst

`successful` The number of successful attempts in the burst

[2]https://github.com/choffa/RTT_app

distance Estimated distance, in millimeters

std_dev The reported standard deviation of the ranging in question

rssi The reported signal strength of the AP at the time of ranging

Each row represents a single burst to a single AP. Furthermore, the name/description, as well as ISO timestamp is recorded in the header section of the `.csv` file.

3.3 Measurements

The actual tests themselves was conducted over two weekends in early May 2020, later than originally planned due to the closure of campus in the wake of the COVID-19 pandemic. As a result there was little to no activity in the building at this time.

A total of 98 sessions, consisting of 30 bursts each, was carried out. These were divided into 16 1D scenarios, ie. using only 1 AP, with 2 sessions in each scenario, as well as 10 2D scenarios, and 12 3D scenarios, with 3 sessions in each. A complete overview of all the sessions can be found in appendix B.

Sessions were set up by placing the AP's over the relevant control point, aligning the point with the center of the AP as best as possible. It is worth noting however, as it is not known exactly where the antenna (and therefore the measuring point) is located within the device, it is impossible to be exact. After the AP was placed in the assigned location, power was was connected and some time was given to allow for complete startup, including connections between the APs as they were set up in a mesh network configuration.

When the status indicators of all included APs indicated that startup had finished, the measurements could commence. A test subject would hold the phone in a natural position standing over the control point and start the ranging. When the application indicated that 30 bursts had been performed, ranging would be ceased manually. Afterwards, the scenario name would be noted, along with the positions of the phone, APs and the time when ranging ended. Then the procedure started over with the next scenario.

3.3.1 Processing

In a post-processing step, the x and y coordinates of the AP was added to the `.csv` for easier data handling down the line. Furthermore, the position of the mobile device was also added to the header section of the file.

For the distance observations, the ranges from the 30 bursts are reduced into one distance estimation using a weighted arithmetic mean

$$\bar{r} = \frac{\sum_i r_i \sigma_i^{-2}}{\sum_i \sigma_i^{-2}} \tag{3.1}$$

In all scenarios the average ranges was then compared to the ground truth using a Matlab script that takes a table corresponding to the `.csv` file as argument `m`, and the ground truth coordinates as a vector `gt`, returning the arithmetic mean of the distance `dist` as well as the difference between this and the ground truth distance `error`. In the 2D and 3D scenario case, `m` should be filtered to only include range estimations towards a single AP.

Listing 3.6: Average distance and error estimation

```
function [dist, error] = calc_distance(m, gt)
    ap_pos = m(1, {'x', 'y'});

    % Remove bad meassurements
    m = m(~isnan(m.distance), :);
    m = m(m.std_dev > 0, :);

    % Calculate observed and ground truth distance
    dist = (m.distance' * m.std_dev.^-2) / sum(m.std_dev.^-2);
    calc_dist = sqrt((gt(1)-ap_pos.x)^2 + (gt(2)-ap_pos.y)^2) *
       1000;

    % Calculate error
    error = dist - calc_dist;
end
```

For the 3D case, the LSM procedure described in 2.3.3 was implemented in another Matlab script. The ranges from each bursts are not reduced before using this function.

Listing 3.7: LSM implementation

```
function [init_xx, xx, S] = LSM(r, x, y, std_dev)
    % Create b vector
    b = r(2:end).^2 - r(1).^2 - (x(2:end).^2 - x(1).^2) - (y(2:
       end).^2 - y(1).^2);

    % Ensure expected form
    if isrow(b)
        b = b';
    end

    % Create design matrix A
    A = 2 .* [x(1) - x(2:end), y(1) - y(2:end)];

    % Initial LSM estimation
    init_xx = (A.'*A)^-1 * A.'*b;

    % Iterative estimation
    i = 0;
    xx = init_xx;
```

```
    delta = [1e3, 1e3];
    W = diag(std_dev.^-2);
    while norm(delta) < 1e6 && norm(delta) > 10 && i < 1e3
        i = i+1;
        f = sqrt((xx(1) - x).^2 + (xx(2) - y).^2);
        J = [(xx(1)-x), (xx(2)-y)] ./ (f);
        K = r - f;
        delta = (J.'*W*J)^-1 * (J.'*W*K);
        xx = xx + delta;
    end

    % Empirical standard deviation calculations
    v = J*xx - K;
    sigma = (v.'*W*v) / (length(K) - length(xx));
    C = (sigma * (J.'*W*J)^-1);
    S = diag(C).^0.5;
end
```

This script returns both the initial position estimation as well as that estimated by the iterative process, in order to compare the two. Note that the iterative process runs until the corrections are less than 10mm, more than 1km, or 1000 iterations have occurred. 10mm was chosen as the lower bound, as it is well within the theoretical accuracy of the system. The latter two was chosen to stop non-converging calculations, since a correction of 1km is clearly unstable at this scale, while 1000 iterations should be more than enough to converge on a solution, should one exist. Finally, the standard deviation of the iterative method is calculated and returned.

Only the LSM procedure was applied to the data. As mentioned, some of the preliminary work suggest using a Kalman filter to improve accuracy further, such as Ciurana et al. (2006). However, this is primarily useful in dynamic scenarios, because of the smoothing and predictive features of such a filter. In fact, they show that a Kalman filter is more accurate than a non linear least squares method based on Newtons method in such a scenario. However, all experiments presented here are static in nature, and therefore Kalman filtering is of little use.

Chapter 4

Results & Discussion

4.1 Results

This section will present the results of the experiments. First, the distance results of the distance estimations is discussed, then the position estimation results. Finally, results will be presented without the weight parameters, to investigate the effectiveness of these.

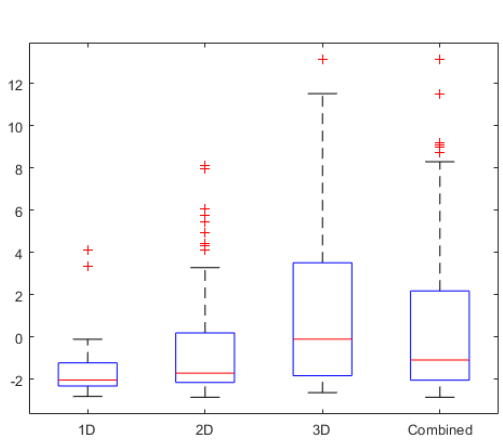
4.1.1 Distance Estimation

	1D	**2D**	**3D**	**Combined**
Mean	1.9096	2.3865	3.0300	2.6550
Minimum	0.0958	0.0107	0.0455	0.0107
Median	2.0643	2.0019	2.0450	2.0376
90th	2.7646	5.1114	8.0244	6.1451
Maximum	4.1441	8.1401	13.1414	13.1414

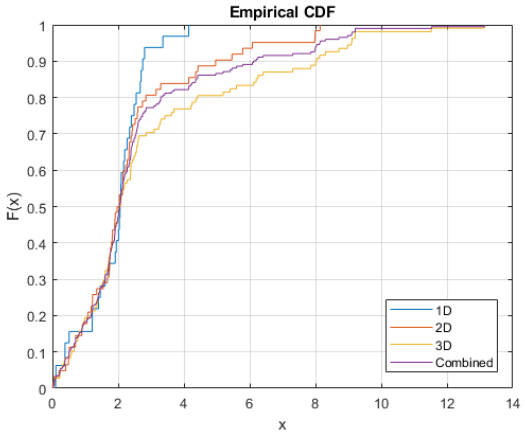
Table 4.1: Absolute Errors of Distance Estimations [m]

Table 4.1 shows the important statistics about the absolute errors of the different scenario types, while complete results are found in appendix C. It clearly shows that the average and median error observed is above 1 meter for all of the scenario types. Furthermore, errors are on average larger in 3D than in 2D, which again is larger than 1D. This is probably due to the increasing likelihood of NLOS propagation, and therefore errors due to multipath, in these scenarios.

One interesting, and slightly confusing result can be seen in the 1D results, where 30 of 32 observations have negative errors. In other words, most of the distance estimations were *shorter* than expected. However, given the basic FTM protocol, the estimated distance should never be significantly to short. Signal propagation speed is fixed, but a distance estimation that is shorter that expected would suggest that the signal arrived sooner than



(a) Boxplot



(b) Cumulative distribution function (Absolute)

Figure 4.1: Plots of Distance Errors

expected, ie. faster than the speed of light. Ibrahim et al. (2018) also encountered this problem, particularly over shorter distances. They suggested that it might be a result of multipath mitigation techniques applied at the hardware or firmware level. There does not appear to be any form of whitepaper that describes the exact implementation of FTM on the Pixel 3a, Google WiFi, or its wireless chips. Therefore, this explanation cannot be ruled out.

Another explanation could be random clock errors. If this was the case, one would assume that the errors are normally distributed $E \sim \mathcal{N}(0, \sigma^2)$, with expectation 0 and unknown variance σ^2. A left tailed t-test can be performed to check this assumption. We test

$$H_0 : \mu = 0$$
$$H_1 : \mu < 0$$

for the 1D session errors. The test statistic is then

$$t = \frac{\bar{E} - \mu}{s/\sqrt{n}} = -5.1913$$

with 31 degrees of freedom. This yields $p = 6.205 \times 10^{-6}$ and H_0 is rejected. It is therefore shown that clock errors cannot be the explanation, and multipath mitigation remains the most likely explanation.

We can also perform a two-tailed t-test on the errors of all sessions combined, that is

$$H_0 : \mu = 0$$
$$H_1 : \mu \neq 0$$

In this case the test statistic $t = 1.9066$ has 201 degrees of freedom and $p = 0.058$. In this case H_0 is not rejected, as we would expect.

4.1.2 Position Estimation

Similar problems with accuracy can be seem in the 3D adjustment results, which can be found in table C.4. First, note that a value of NaN suggest that there was insufficient successful measurements to at least one AP. This might happen when an AP is within range of the WiFi scan, but the signal is not strong enough to reliably perform FTM range estimation. These results are discussed more in detail in section 4.2.1.

Second, we observe larger errors than expected across the board, but especially in sessions 3D-1 through 3D-6, which are clearly unstable. This can also be shown by increasing the maximum allowed correction in listing 3.7, which causes a similar increase in the error. One possible explanation is that these scenarios might be more prone to multipath or poor geometry. This is further discussed in 4.2.2. However, these 6 sessions will be considered outliers in further discussions.

	With outliers		**Without outliers**	
	Initial	**Iterative**	**Initial**	**Iterative**
Mean	189.8990	304.9793	14.8937	6.8462
Minimum	4.0768	1.4908	4.0768	1.4908
Median	13.1885	7.5304	9.9851	6.7028
90th	866.7214	1504.7491	43.3733	10.7996
Maximum	1564.3748	1728.1652	50.0391	12.7218

Table 4.2: Errors of Position Estimates [m]

Table 4.2 shows the various statistics for the errors of the two methods, with and without the previously discussed unstable outliers. The errors are defined as the Euclidean distance from the estimation to the ground truth point. We observe that, as we would expect, the iterative process reduces the errors and improves accuracy. In fact the error is reduced in 19 of 24 sessions, with an average improvement of 10.6 meters in these cases, with an average improvement of 8 meters overall. However, it is important to note that, while accuracy improves, it is still far worse than the expected one meter accuracy.

	S_x	S_y
Mean	11.7403	39.2251
Minimum	4.6904	7.9755
Median	8.6703	22.3042
Maximum	33.3184	215.2045

Table 4.3: Standard deviations of Position Estimates

As for the standard deviations, shown in table 4.3 excluding outliers, they are somewhat high, but justifiable given the observed errors. However, it does indicate that the

precision of the system is low.

Also, we note that the standard deviations are significantly higher in the y direction. This can easily be explained by the geometry of the sessions. The most heavily used points are P1-19 through P1-23, as well as P2-05 trough P2-09. As can be seen in appendix A, these points have significantly more spread in the x direction, left to right on the figure, than in the y direction, top to bottom on the figure.

4.1.3 Results Without Weighting

	Un-Weighted	**Difference**
Mean	6.0952	-0.7510
Minimum	1.6132	0.1224
Median	5.3624	-1.3404
90th	10.2940	-0.5056
Maximum	17.6852	4.9633

Table 4.4: Errors of Un-Weighted Position Estimates [m]

For comparisons sake, the position estimations was calculated without the weight parameter. The results of this can be found in table C.5, and major statistics can be found in table 4.4. Some interesting observations can be gleaned from here, then mean improves by 75cm, while the median shows a full 1.3 meters of improvement when compared to the weighted results. However, the change in mean does not reach statistical significance when a two-sample t-test is performed

$$t = \frac{\hat{E}_w - \hat{E}_u}{\sqrt{\frac{S_w^2}{n} + \frac{S_u^2}{n}}} = 0.7891$$

where $\hat{E}_w$ and $\hat{E}_u$ is mean error of weighted and un-weighted adjustments respectively. This gives 43.0727 degrees of freedom and $p = 0.4344$. Thus we cannot conclude that adjustments without weights are better, nor the other way around.

4.2 Discussion

Clearly some of the results in this survey are not quite what we would expect. Errors are large and sometimes surprisingly so. However, there are potential sources of errors. For example, the previously mentioned multipath mitigation, or bad geometry. Though, less than optimal geometry might make for more realistic scenarios, as WiFi APs are rarely deployed in such a manner as to take into account the optimal geometry from a geomatics standpoint.

Evidence of less than optimal geometry can also be found in the increased standard deviation of the y coordinate. However, it is more difficult to argue that this represents a more realistic setup. The survey was constrained to the control points that where already established. Furthermore, points in the area with the highest density where no longer available as they had been destroyed by the renovation work. Therefore, the survey was limited to relatively low density areas and, more importantly, mainly corridors, where control points would closely align with each other. In an actual deployment, some APs should be available in some of the study halls and offices that line the corridor, ensuring a better spread in both directions.

Furthermore, the height difference of the phone and the APs should result in a small error in the distance estimations, causing the estimations to be biased towards overshooting. However, this error will decrease with distance.

In general the ranging errors are about on par with that is observed by Ibrahim et al. (2018), but higher than those observed by the other papers presented in section 2.4. A possible explanation could be differences in geometry. Šeleng (2019) for example, only tests distances between 1 and 17 meters, with most tests being below 10 meters in distance. The average ground truth distance in this study is almost 20 meters. The location errors, on the other hand, appear to be worse than all of them.

4.2.1 Invalid Results

By comparing the position results that returned NaN with the individual distance estimations in these sessions, we can begin to understand why these failed. There are two specific failure conditions that arise, that should be discussed individually.

Session	**P1-06**	**P2-01**	**P2-04**
3D-7	4.2105	8.7127	NaN
3D-8	-0.2372	6.1988	NaN
3D-9	-0.2310	5.1893	NaN

Table 4.5: Distance Errors of First Failure Condition

First is the failure found in scenarios 3D-7, 3D-8, and 3D-9, which errors are given in table 4.5. In this scenario, APs were spread across two floors; P1-06, P2-01, and P2-04, while the smartphone was placed in P1-09. This means that two of the signals had to travel trough the concrete floor, something that the signal from P2-04 clearly fails to do.

Also observe the relatively large errors in the estimation to P2-01. Again this makes sense, since the signal must travel trough the floor, but also because only the 2D coordinates of each point is available, while most of the distance in this case comes from the height component. This is not to say that using APs across floors is completely useless.

Session	Error
3D-22	1.3897
3D-23	0.7991
3D-24	1.4486
3D-25	0.0455
3D-26	1.2998
3D-27	0.0711

Table 4.6: Errors of APs on Different Floor

We can compare these results to sessions 3D-22 trough 3D-27, which all uses one AP on a different floor than the target. The errors of the distance across floors is shown in table 4.6. The error is quite acceptable, with a mean of 0.84 meters. However, it should be noted that these measurements where conducted in a more open space, next to an open stairwell, resulting in more or less LOS to the AP.

Figure 4.2: Setups of Second Failure Condition

The second failure condition is found in sessions 3D-22, 3D-24, 3D-26, 3D-29, 3D-31, and 3D-33. The setups of these scenarios are found in figure 4.2, where purple (solid) is the setup of 3D-22, 3D-24, and 2D-26, while green (dotted) is the setup of the other three.

The specific failure in this instance comes from P1-23. All sessions fail to obtain a usable measurement. This could be explained by the many walls that are in the LOS path of the signal. While this is one of the harder multipath problems in the study, others that seem similiar in complexity still obtain usable measurements. However, the consistency of failure over these 6 sessions seem to indicate that this scenario is somehow more complex.

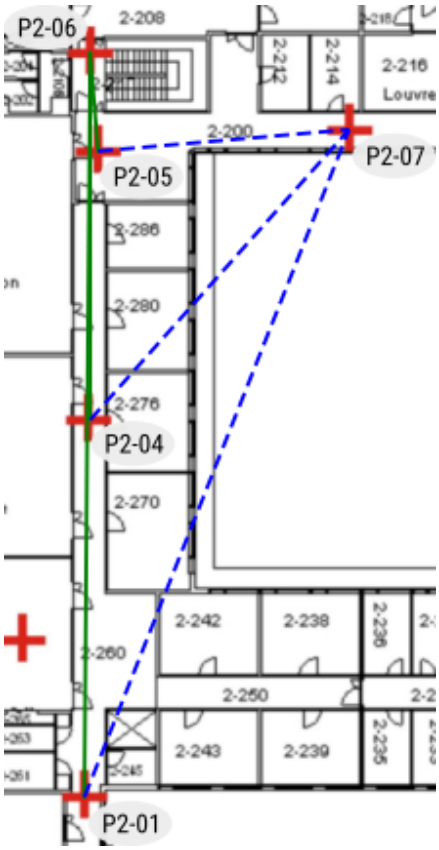


Figure 4.3: Setup of Outlier Sessions

4.2.2 Outliers

The results from sessions 3D-1 to 3D-6 are unstable. This might be a result of a particularly poor geometry in these sessions or the related multipath problem. The setups are shown in figure 4.3, where blue (dotted) is the setup of 3D-1, 3D-3, and 3D-5, while green (solid) is the setups of 3D-2, 3D-4, and 3D-6.

Session		**P2-01**	**P2-04**	**P2-05**
Blue (Dotted)	3D-1	9.2104	8.0335	-1.7587
	3D-3	9.2115	7.8608	-1.5412
	3D-5	9.1122	8.1242	-1.6007
Green (Solid)	3D-2	-1.7856	-2.4311	-0.8188
	3D-4	-1.7051	-2.2750	-1.6954
	3D-6	-1.8855	-2.6191	-1.9060

Table 4.7: Outlier Distance Errors

Starting with the blue (dotted) scenario, there might be some problems with suboptimal geometry, as all APs more or less align. Furthermore, this is a hard multipath scenario, especially since the large empty area in the middle of the map is in fact an open sky atrium, so in LOS propagation the signal would have to travel trough two outer

walls. This is unlikely. It is more likely that the signal followed the corridor. This is also reflected in the errors of the distance estimations, as shown in table 4.7. We see that those two points, P2-01 and P2-04, have particularly large errors. This would explain why the LSM adjustment was unstable.

Moving on to the green (solid) scenario, observe that the measured point now also aligns with the AP points, further exacerbating the poor geometry. Another interesting observation is that all of the distance errors are negative. In fact, this is the only sessions where this is the case. This is likely to have caused the LSM estimation to have no solution.

It is hard to see how problems like this can be avoided without already knowing the geometry of the problem, as would be the case in normal pedestrian navigation. However, again the introduction of more APs could help, as this would make the LSM estimation more robust to outliers. Another possibility would be to filter/weight on RSSI, as this might give some indication to the "multipathness" of the current measurements, or otherwise include RSSI in the estimation, as proposed by Guo et al. (2019).

Chapter 5

Conclusion

In general, this survey does not find compelling evidence for 1 meter accuracy using FTM with Google Pixel 3a and Google WiFi in an office environment. In fact, median distance errors where double that at about 2 meters. Position estimation resulted in even higher errors, over 6 meters median. While some weaknesses of this study have been discussed, they do not explain the higher than expected errors observed, especially the bias towards underestimating the distance in simple LOS scenarios.

Future studies should investigate other platforms that comply to the standard, as well as investigating the impacts of denser AP deployment. It would also be interesting to test how this specific configuration performs in a more dynamic scenario where the user is moving.

It is also shown that weighting does not significantly improve the results. Future work could also test this on a larger scale, as well as look into the effects of other possible weights.

Accuracy might also be further improved by employing fusion methods that combine FTM with some other, more established method, as we already see some examples of in literature.

Nevertheless, FTM is still a good candidate for indoor positioning. Even a minimal deployment can achieve results that are usable at least in some contexts. And while some fingerprinting systems have been shown to perform better, they still require the offline survey step. This viability is also clear in the literature about FTM as well as time based ranging using WiFi in general.

Bibliography

E. Au. The latest progress on IEEE 802.11mc and IEEE 802.11ai [standards]. *IEEE Vehicular Technology Magazine*, 11(3):19–21, 2016. doi: 10.1109/MVT.2016.2586398.

P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *19th Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 775–784. IEEE, 2000. doi: 0.1109/INFCOM.2000.832252.

B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer. EasyLiving: Technologies of interlligent environments. In P. Thomas and G. Hans-W., editors, *Handheld and Ubiquitous Computing*, volume 1927 of *Lecture Notes in Computer Science*. Springer, 2002. doi: 10.1007/3-540-39959-3_2.

I. Casacuberta and A. Ramirez. Time-of-flight positioning using the existing wireless local area network infrastructure. In *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2012. doi: 10.1109/IPIN.2012.6418938.

G. Caso, M. T. P. Le, L. De Nardis, and M.-G. Di Benedetto. Performance comparison of WiFi and UWB fingerprinting indoor positioning systems. *Technologies*, 6(1):14, 2018. doi: 10.3390/technologies6010014.

J. Chen and K. C. Clarke. Indoor cartography. *Cartography and Geographic Information Science*, 2019. doi: 10.1080/15230406.2019.1619482.

M. Ciurana, F. Barceló, and S. Cugno. Indoor tracking in WLAN location with TOA measurements. In *Proceedings of the 4th ACM international workshop on mobility management and wireless access*, MobiWac '06, pages 121–125. ACM, 2006. doi: 10.1145/1164783.1164806.

M. Ciurana, D. López, and F. Barceló-Arroyo. SofTOA: Software ranging for TOA-based positioning of WLAN terminals. In T. Choudhurt, A. Quigley, T. Strang, and K. Suginuma, editors, *Location and Context Awereness*, volume 5561 of *Lecture Notes in Computer Science*, pages 207–221, Berlin, 2009. Springer. doi: 10.1007/978-3-642-01721-6_13.

M. Ciurana, F. Barceló-Arroyo, and I. Martín-Escalona. Comparative performance evaluation of IEEE 802.11v for positioning with time of arrival. *Computer Standards and Interfaces*, 33 (3):344–349, 2011. doi: 10.1016/j.csi.2010.09.003.

M. Cominelli, P. Patras, and F. Gringoli. Dead on arrival: An empirical study of the bluetooth 5.1 positioning system. In *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, WiNTECH '19. ACM Press, 2019. doi: 10.1145/3349623.3355475.

W. Dargie and C. Poellabauer. *Fundamentals of Wireless Sensor Networks: Theory and Practice*. Wiley Series on Wireless Communications and Mobile Computing. Wiley, Chichester, 2010.

C. D. Ghilani. *Adjustment Computations: Spatial Data Analysis*. Wiley, Hoboken, NJ, 5th edition, 2010.

A. Gunther and C. Hoene. Measuring round trip times to determine the distance between WLAN nodes. In R. Boutaba, K. Almeroth, R. Puigjaner, S. Shen, and J. P. Black, editors, *Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, volume 3462 of *Lecture Notes in Computer Science*, pages 768–779, Berlin, 2005. Springer. doi: 10.1007/11422778_62.

G. Guo, R. Chen, F. Ye, X. Peng, Z. Liu, and Y. Pan. Indoor smartphone localization: A hybrid WiFi RTT-RSS ranging approach. *IEEE Access*, 7:176767–176781, 2019. doi: 10.1109/ACCESS.2019.2957753.

R. K. Harle and A. Hopper. Deploying and evaluating a location-aware system. In *3rd International Conference on Mobile Systems, Applications, and Services*, MobiSys '05, page 219–232, New York, NY, USA, 2005. Association for Computing Machinery. doi: 10.1145/1067170.1067194.

C. Hoene and J. Willmann. Four-way toa and software-based trilateration of IEEE 802.11 devices. In *19th International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1–6. IEEE, 2008. doi: 10.1109/PIMRC.2008.4699394.

M. Ibrahim, H. Liu, M. Jawahar, V. Nguyen, M. Gruteser, R. Howard, B. Yu, and F. Bai. Verification: Accuracy evaluation of WiFi fine time measurements on and open platform. In *MobiCom '18: The 24th Annual International Conference on Mobile Computing and Networking*, pages 417–427. Association for Computing Machinery, 2018. doi: 10.1145/3241539.3241555.

IEEE. Standard for information technology—telecommunications and information exchange between systems local and metropolitan area networks—specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, 2016. doi: 10.1109/IEEESTD.2016.7786995.

F. Izquierdo, M. Ciurana, F. Barceló, J. Paradells, and E. Zola. Performance evaluation of a toa-based trilateration method to locate terminals in WLAN. In *1st International Symposium on Wireless Pervasive Computing*, volume 2006, pages 1–6. IEEE, 2006. doi: 10.1109/ISWPC.2006.1613598.

B. Jang and H. Kim. Indoor positioning technologies without offline fingerprinting map: A survey. *IEEE Communications Surveys and Tutorials*, 21(1):508–525, 2019. doi: 10.1109/COMST.2018.2867935.

S. König, M. Schmidt, and C. Hoene. Precise time of flight measurements in IEEE 802.11 networks by cross-correlating the sampled signal with a continuous barker code. In *7th IEEE International Conference on Mobile ad-hoc and Sensor Systems*, pages 642–649. IEEE, 2010. doi: 10.1109/MASS.2010.5663785.

H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, 2007. doi: 10.1109/TSMCC.2007.905750.

K. Liu, X. Liu, and X. Li. Guoguo: Enabling fine-grained smartphone localization via acoustic anchors. *IEEE Transactions on Mobile Computing*, 15(5):1144–1156, 2016. doi: 10.1109/TMC.2015.2451628.

G. Lu, Y. Yan, L. Ren, P. Saponaro, N. Sebe, and C. Kambhamettu. Where am i in the dark: Exploring active transfer learning on the use of indoor localization based on thermal imaging. *Neurocomputing*, 173:83 – 92, 2016. doi: https://doi.org/10.1016/j.neucom.2015.07.106.

A. Makki, A. Siddig, M. Saad, and C. Bleakley. Survey of WiFi positioning using time-based techniques. *Computer Networks*, 88:218–233, 2015. doi: 10.1016/j.comnet.2015.06.015.

A. Mandal, C. V. Lopes, T. Givargis, A. Haghighat, R. Jurdak, and P. Baldi. Beep: 3d indoor positioning using audible sound. In *Second IEEE Consumer Communications and Networking Conference, 2005. CCNC. 2005*, pages 348–353, 2005. doi: 10.1109/CCNC.2005.1405195.

T. Midtbø, A. S. Nossum, T. A. Haakonsen, and R. P. V. Nordan. Are indoor positioning systems mature for cartographic tasks?, 2012.

L. M. Ni, Y. Liu, C. L. Yiu, and A. P. Patil. Landmarc: Indoor location sensing using active rfid. *Wireless Networks*, 10(6):701–710, 11 2004. doi: 10.1023/B:WINE.0000044029.06344.dd.

C. Peng, G. Shen, and Y. Zhang. BeepBeep: A high-accuracy acoustic-based system for ranging and localization using cots devices. *ACM Transactions on Embedded Computing Systems*, 11 (1), 2012. doi: 10.1145/2146417.2146421.

N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. In *6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, pages 32–43. ACM, 2000.

S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, Harlow, England, 3rd edition, 2016.

L. Schauer, F. Dormeister, and M. Maier. Potentials and limitations of WiFi-positioning using time-of-flight. In *International Conference on Indoor Positioning and indoor Navigation*, pages 1–9. IEEE, 2013. doi: 10.1109/IPIN.2013.6817861.

W. Shao, H. Luo, F. Zhao, Y. Ma, Z. Zhao, and A. Crivello. Indoor positioning based on fingerprint-image and deep learning. *IEEE Access*, 6:74699–74712, 2018. doi: 10.1109/ACCESS.2018.2884193.

A. A. N. Shirehjini, A. Yassine, and S. Shirmohammadi. An rfid-based position and orientation measurement system for mobile objects in intelligent environments. *IEEE Transactions on Instrumentation and Measurement*, 61(6):1664–1675, 2012. doi: 10.1109/TIM.2011.2181912.

M. Uradzinski, H. Guo, X. Liu, and M. Yu. Advanced indoor positioning using zigbee wireless technology. *Wireless Personal Communications*, 97(4):6509–6518, 2017. doi: 10.1007/s11277-017-4852-5.

D. Vasisht, S. Kumar, and D. Katabi. Decimeter-level localization with a single wifi access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 165–178, Santa Clara, CA, Mar. 2016. USENIX Association.

R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10(1):91–102, Jan. 1992. doi: 10.1145/128756.128759.

WiFi Alliance. Wi-Fi CERTIFIED Location brings Wi-Fi indoor positioning capabilities, 2017. URL `https://goo.gl/BSUCdG`.

S. Xu, R. Chen, Y. Yu, G. Guo, and L. Huang. Locating smartphones indoors using built-in sensors and Wi-Fi ranging with an enhanced particle filter. *IEEE Access*, 7:95140–95153, 2019. doi: 10.1109/ACCESS.2019.2927387.

G. Yanying, A. Lo, and I. Niemegeers. A survey of indoor positioning systems for wireless personal networks. *IEEE Communications Surveys and Tutorials*, 11(1):13–32, 2009. doi: 10.1109/SURV.2009.090103.

M. Youssef and A. Agrawala. The horus location determination system. *Wireless Networks*, 14 (3):357–374, 2008. doi: 10.1007/s11276-006-0725-7.

Y. Yu, R. Chen, L. Chen, G. Guo, F. Ye, and Z. Liu. A robust dead reckoning algorithm based on Wi-Fi FTM and multiple sensors. *Remote Sensing*, 11(5):504–526, 2019. doi: 10.3390/rs11050504.

F. Zafari. *iBeacon based proximity and indoor localization system.* Master thesis, Purdue University, 2016. URL `https://docs.lib.purdue.edu/open_access_theses/767`.

F. Zafari, A. Gkelias, and K. K. Leung. A survey of indoor localization systems and technologies. *IEEE Communications Surveys and Tutorials*, 21(3):2568–2599, 2019. doi: 10.1109/COMST.2019.2911558.

C. Zhang, K. P. Subbu, J. Luo, and J. Wu. Groping: Geomagnetism and crowdsensing powered indoor navigation. *IEEE Transactions on Mobile Computing*, 14(2):387–400, 2015. doi: 10.1109/TMC.2014.2319824.

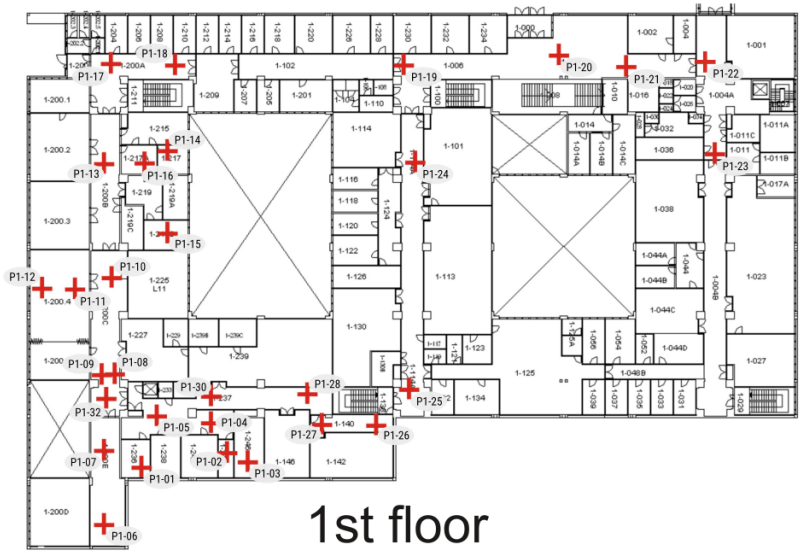
L. Zhu, A. Yang, D. Wu, and L. Liu. Survey of indoor positioning technologies and systems. In S. Ma, L. Jia, X. Li, L. Wang, H. Zhou, and X. Sun, editors, *International Conference on Life System Modeling and Simulation*, volume 461 of *Communications in Computer and Information Science*, pages 400–409. Springer, 2014. doi: 10.1007/9783662452837_41.

M. Šeleng. *WiFi Round Trip Time for Indoor Navigation.* Master thesis, Masaryk University, 2019.

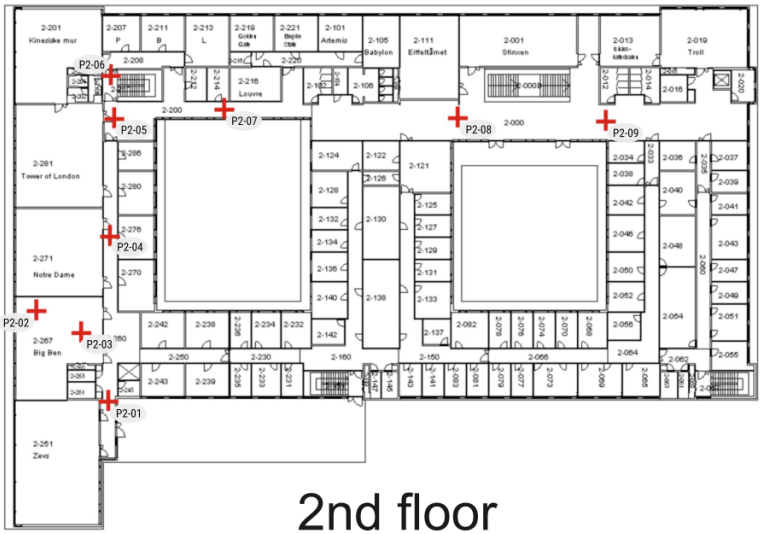
Appendices

Appendix A

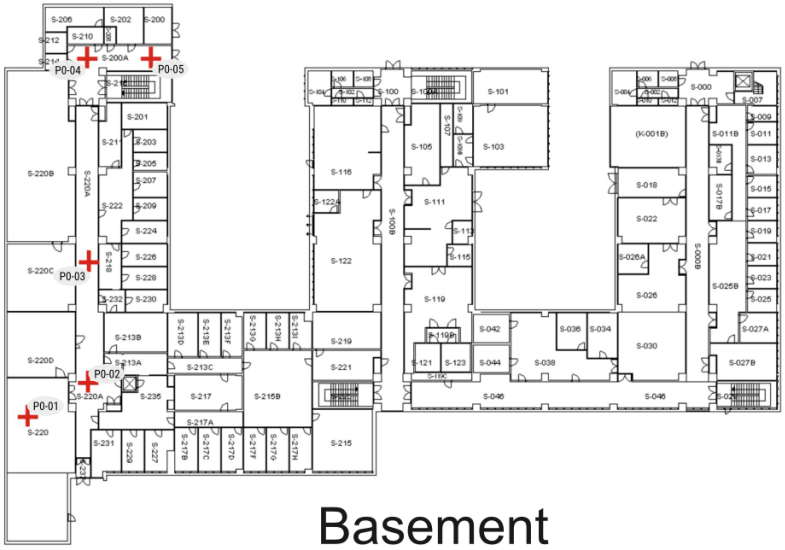
Indoor control points



1st floor



2nd floor



Basement

Name	x	y
P0-01	5.91	118.32
P0-02	14.02	111.45
P0-03	13.05	99.19
P0-04	13.03	74.14
P0-05	21.20	74.13
P1-01	18.31	121.55
P1-02	28.37	119.97
P1-03	29.94	120.81
P1-04	25.18	116.43
P1-05	19.54	115.03
P1-06	12.75	129.38
P1-07	12.75	119.92
P1-08	14.07	110.46
P1-09	13.06	110.46
P1-10	13.91	99.39
P1-11	9.66	99.26
P1-12	5.41	99.27
P1-13	13.05	86.77
P1-14	19.92	83.69
P1-15	20.56	94.35
P1-16	16.79	85.27
P1-17	13.34	73.64
P1-18	20.76	74.15
P1-19	48.47	74.12
P1-20	62.22	73.99
P1-21	73.26	73.95
P1-22	84.63	73.63
P1-23	85.48	85.10
P1-24	49.54	92.69
P1-25	48.84	112.21
P1-26	45.20	116.19
P1-27	38.18	117.08
P1-28	36.60	112.73
P1-30	25.72	113.78
P1-31	31.79	110.08
P1-32	12.54	113.57
P2-01	15.98	117.20
P2-02	6.85	103.97
P2-03	13.45	107.97
P2-04	16.63	95.73
P2-05	17.12	81.47
P2-06	16.03	76.10
P2-07	30.72	80.51
P2-08	58.64	81.27
P2-09	76.97	81.28

Table A.1: Point Coordinates

Appendix B

Scenarios

Name	Phone	AP1	AP2	AP3
1D-01	P2-08	P2-09	-	-
1D-02	P2-07	P2-09	-	-
1D-03	P2-08	P2-09	-	-
1D-04	P2-07	P2-09	-	-
1D-05	P2-05	P2-06	-	-
1D-06	P2-07	P2-06	-	-
1D-07	P2-05	P2-06	-	-
1D-08	P2-07	P2-06	-	-
1D-09	P2-06	P2-07	-	-
1D-10	P2-05	P2-07	-	-
1D-11	P2-06	P2-07	-	-
1D-12	P2-05	P2-07	-	-
1D-13	P2-08	P2-07	-	-
1D-14	P2-09	P2-07	-	-
1D-15	P2-08	P2-07	-	-
1D-16	P2-09	P2-07	-	-
1D-17	P1-20	P1-21	-	-
1D-18	P1-19	P1-21	-	-
1D-19	P1-20	P1-21	-	-
1D-20	P1-19	P1-21	-	-
1D-21	P1-22	P1-21	-	-
1D-22	P1-23	P1-21	-	-
1D-23	P1-22	P1-21	-	-
1D-24	P1-23	P1-21	-	-
1D-25	P1-19	P1-20	-	-
1D-26	P1-21	P1-20	-	-
1D-27	P1-19	P1-20	-	-
1D-28	P1-21	P1-20	-	-
1D-29	P1-22	P1-20	-	-
1D-30	P1-23	P1-20	-	-
1D-31	P1-22	P1-20	-	-
1D-32	P1-23	P1-20	-	-

Table B.1: 1D Experiment setups

Name	Phone	AP1	AP2	AP3
2D-01	P1-20	-	P1-24	P1-21
2D-02	P1-20	P1-19	-	P1-21
2D-03	P1-20	-	P1-24	P1-21
2D-04	P1-20	P1-19	-	P1-21
2D-05	P1-20	-	P1-24	P1-21
2D-06	P1-20	P1-19	-	P1-21
2D-07	P1-21	P1-20	P1-23	-
2D-08	P1-22	P1-20	P1-23	-
2D-09	P1-21	P1-20	P1-23	-
2D-10	P1-22	P1-20	P1-23	-
2D-11	P1-21	P1-20	P1-23	-
2D-12	P1-22	P1-20	P1-23	-
2D-13	P1-24	P1-19	P1-25	-
2D-14	P1-24	P1-19	P1-25	-
2D-15	P1-24	P1-19	P1-25	-
2D-16	P2-08	P2-05	P2-09	-
2D-17	P2-07	P2-05	P2-09	-
2D-18	P2-08	P2-05	P2-09	-
2D-19	P2-07	P2-05	P2-09	-
2D-20	P2-08	P2-05	P2-09	-
2D-21	P2-07	P2-05	P2-09	-
2D-22	P2-07	P2-05	P2-04	-
2D-23	P2-06	P2-05	P2-04	-
2D-24	P2-07	P2-05	P2-04	-
2D-25	P2-06	P2-05	P2-04	-
2D-26	P2-07	P2-05	P2-04	-
2D-27	P2-06	P2-05	P2-04	-
2D-28	P2-05	P2-04	P2-01	-
2D-29	P2-05	P2-04	P2-01	-
2D-30	P2-05	P2-04	P2-01	-

Table B.2: 2D Experiment setups

Name	Phone	AP1	AP2	AP3
3D-01	P2-07	P2-05	P2-04	P2-01
3D-02	P2-06	P2-05	P2-04	P2-01
3D-03	P2-07	P2-05	P2-04	P2-01
3D-04	P2-06	P2-05	P2-04	P2-01
3D-05	P2-07	P2-05	P2-04	P2-01
3D-06	P2-06	P2-05	P2-04	P2-01
3D-07	P1-09	P1-06	P2-04	P2-01
3D-08	P1-09	P1-06	P2-04	P2-01
3D-09	P1-09	P1-06	P2-04	P2-01
3D-10	P2-07	P2-08	P2-05	P2-06
3D-11	P2-07	P2-08	P2-05	P2-06
3D-12	P2-07	P2-08	P2-05	P2-06
3D-13	P2-09	P2-08	P2-05	P2-06
3D-14	P2-09	P2-08	P2-05	P2-06
3D-15	P2-09	P2-08	P2-05	P2-06
3D-16	P2-07	P2-08	P2-05	P2-04
3D-17	P2-07	P2-08	P2-05	P2-04
3D-18	P2-07	P2-08	P2-05	P2-04
3D-19	P2-07	P2-08	P2-06	P2-04
3D-20	P2-07	P2-08	P2-06	P2-04
3D-21	P2-07	P2-08	P2-06	P2-04
3D-22	P1-20	P1-19	P2-08	P1-23
3D-23	P1-21	P1-19	P2-08	P1-23
3D-24	P1-20	P1-19	P2-08	P1-23
3D-25	P1-21	P1-19	P2-08	P1-23
3D-26	P1-20	P1-19	P2-08	P1-23
3D-27	P1-21	P1-19	P2-08	P1-23
3D-28	P1-21	P1-19	P1-22	P1-23
3D-29	P1-20	P1-19	P1-22	P1-23
3D-30	P1-21	P1-19	P1-22	P1-23
3D-31	P1-20	P1-19	P1-22	P1-23
3D-32	P1-21	P1-19	P1-22	P1-23
3D-33	P1-20	P1-19	P1-22	P1-23
3D-34	P1-20	P1-19	P1-22	P1-24
3D-35	P1-20	P1-19	P1-22	P1-24
3D-36	P1-20	P1-19	P1-22	P1-24
3D-37	P1-20	P1-19	P1-22	P1-23
3D-38	P1-20	P1-19	P1-22	P1-23
3D-39	P1-20	P1-19	P1-22	P1-23

Table B.3: 3D Experiment setups

Appendix C

Complete Results

Session	Distance	Error
1D-1	17.1156	-1.2144
1D-2	44.0593	-2.1971
1D-3	16.2588	-2.0712
1D-4	43.9099	-2.3466
1D-5	3.8726	-1.6069
1D-6	15.2419	-0.0958
1D-7	3.4654	-2.0141
1D-8	15.2334	-0.1043
1D-9	14.9559	-0.3817
1D-10	11.5968	-2.0370
1D-11	14.9592	-0.3784
1D-12	11.6915	-1.9423
1D-13	25.1263	-2.8041
1D-14	44.7993	-1.4571
1D-15	25.2341	-2.6963
1D-16	44.5422	-1.7142
1D-17	8.7708	-2.2693
1D-18	22.8799	-1.9107
1D-19	8.8748	-2.1653
1D-20	23.3888	-1.4018
1D-21	9.3172	-2.0573
1D-22	16.0343	-0.5081
1D-23	9.2947	-2.0798
1D-24	15.3321	-1.2103
1D-25	11.0029	-2.7477
1D-26	8.4898	-2.5503
1D-27	11.0816	-2.6690
1D-28	8.5577	-2.4824
1D-29	20.3211	-2.0918
1D-30	29.1359	3.3588
1D-31	20.0134	-2.3995
1D-32	29.9213	4.1441

Table C.1: 1D session results

Session	Distance 1	Error 1	Distance 2	Error 2
2D-1	8.7151	-2.3250	27.0177	4.4240
2D-2	8.4413	-2.5988	13.2388	-0.5118
2D-3	9.1473	-1.8927	27.5557	4.9621
2D-4	8.5227	-2.5174	12.8413	-0.9093
2D-5	9.0042	-2.0359	26.7369	4.1433
2D-6	8.6009	-2.4392	13.2633	-0.4873
2D-7	20.8971	4.3547	8.1983	-2.8418
2D-8	22.0022	5.4598	8.3032	-2.7369
2D-9	10.2779	-1.2235	20.2704	-2.1425
2D-10	22.3290	5.7866	9.0722	-1.9679
2D-11	10.3221	-1.1794	20.6706	-1.7423
2D-12	22.6216	6.0792	9.1295	-1.9105
2D-13	10.1544	-1.3471	20.5941	-1.8188
2D-14	20.5794	1.0468	16.7747	-1.8261
2D-15	22.6875	3.1550	16.5626	-2.0382
2D-16	19.7336	0.2011	16.8394	-1.7614
2D-17	17.8261	-0.5039	41.5312	0.0107
2D-18	43.8643	-2.3922	12.9324	-0.7015
2D-19	19.5493	1.2193	41.1172	-0.4033
2D-20	44.1081	-2.1483	14.7105	1.0767
2D-21	21.6226	3.2926	40.8166	-0.7039
2D-22	43.9156	-2.3408	12.7366	-0.8972
2D-23	28.7343	7.9936	11.8802	-1.7536
2D-24	17.3659	-2.2733	3.5795	-1.9000
2D-25	28.7203	7.9796	12.0810	-1.5528
2D-26	17.0577	-2.5814	3.6861	-1.7934
2D-27	28.8808	8.1401	11.4889	-2.1449
2D-28	17.5077	-2.1315	5.5333	0.0538
2D-29	19.0624	-2.4175	34.0736	-1.6745
2D-30	19.2554	-2.2244	34.0088	-1.7394
2D-31	19.1510	-2.3288	34.0275	-1.7207

Table C.2: 2D session results

Session	Distance 1	Error 1	Distance 2	Error 2	Distance 3	Error 3
3D-1	48.7506	9.2104	28.7742	8.0335	11.8751	-1.7587
3D-2	39.3144	-1.7856	17.2081	-2.4311	4.6607	-0.8188
3D-3	48.7517	9.2115	28.6015	7.8608	12.0926	-1.5412
3D-4	39.3949	-1.7051	17.3642	-2.2750	3.7841	-1.6954
3D-5	48.6524	9.1122	28.8649	8.1242	12.0331	-1.6007
3D-6	39.2145	-1.8855	17.0201	-2.6191	3.5735	-1.9060
3D-7	16.0580	8.7127	23.1330	4.2105	NaN	NaN
3D-8	13.5441	6.1988	NaN	NaN	18.6854	-0.2372
3D-9	12.5346	5.1893	18.6915	-0.2310	NaN	NaN
3D-10	17.9319	2.5942	12.0473	-1.5865	27.3497	-0.5807
3D-11	15.6901	0.3524	12.6553	-0.9785	27.5549	-0.3754
3D-12	17.8940	2.5564	12.1232	-1.5106	27.0081	-0.9223
3D-13	64.0123	2.8526	59.0830	-0.7673	16.4262	-1.9038
3D-14	63.7672	2.6075	59.1881	-0.6623	16.7289	-1.6011
3D-15	64.2531	3.0933	59.1454	-0.7049	16.3921	-1.9379
3D-16	29.0423	8.3016	11.8944	-1.7394	27.9894	0.0590
3D-17	28.7436	8.0029	12.6882	-0.9457	31.6229	3.6926
3D-18	29.8385	9.0978	13.3331	-0.3007	27.3794	-0.5509
3D-19	32.2667	11.5260	17.5144	2.1767	26.2564	-1.6739
3D-20	33.8821	13.1414	17.1490	1.8113	26.5449	-1.3854
3D-21	29.7165	8.9758	16.5010	1.1633	26.5521	-1.3783
3D-22	9.5023	1.3897	11.7640	-1.9866	NaN	NaN
3D-23	18.7351	2.1927	17.1493	0.7991	22.4156	-2.3750
3D-24	9.5612	1.4486	12.8588	-0.8918	NaN	NaN
3D-25	19.8119	3.2695	16.3956	0.0455	22.4097	-2.3809
3D-26	9.4124	1.2998	11.7383	-2.0123	NaN	NaN
3D-27	19.9610	3.4186	16.4212	0.0711	22.6962	-2.0944
3D-28	20.1624	3.6200	8.9883	-2.3863	26.2256	1.4350
3D-29	20.4033	-2.0096	11.7344	-2.0162	NaN	NaN
3D-30	22.1382	5.5958	8.8103	-2.5642	30.1926	5.4020
3D-31	20.4257	-1.9872	11.8884	-1.8623	NaN	NaN
3D-32	19.8384	3.2960	8.8516	-2.5229	26.5067	1.7161
3D-33	20.5201	-1.8928	11.6573	-2.0933	NaN	NaN
3D-34	25.8284	3.2348	23.0689	0.6560	11.3743	-2.3763
3D-35	27.0109	4.4172	29.7248	7.3119	11.2717	-2.4789
3D-36	26.8454	4.2518	26.7708	4.3579	11.3614	-2.3892
3D-37	31.8993	6.1222	20.3347	-2.0782	13.9768	0.2262
3D-38	32.0284	6.2512	20.3391	-2.0738	14.9441	1.1935
3D-39	32.1856	6.4085	20.2877	-2.1252	15.1013	1.3507

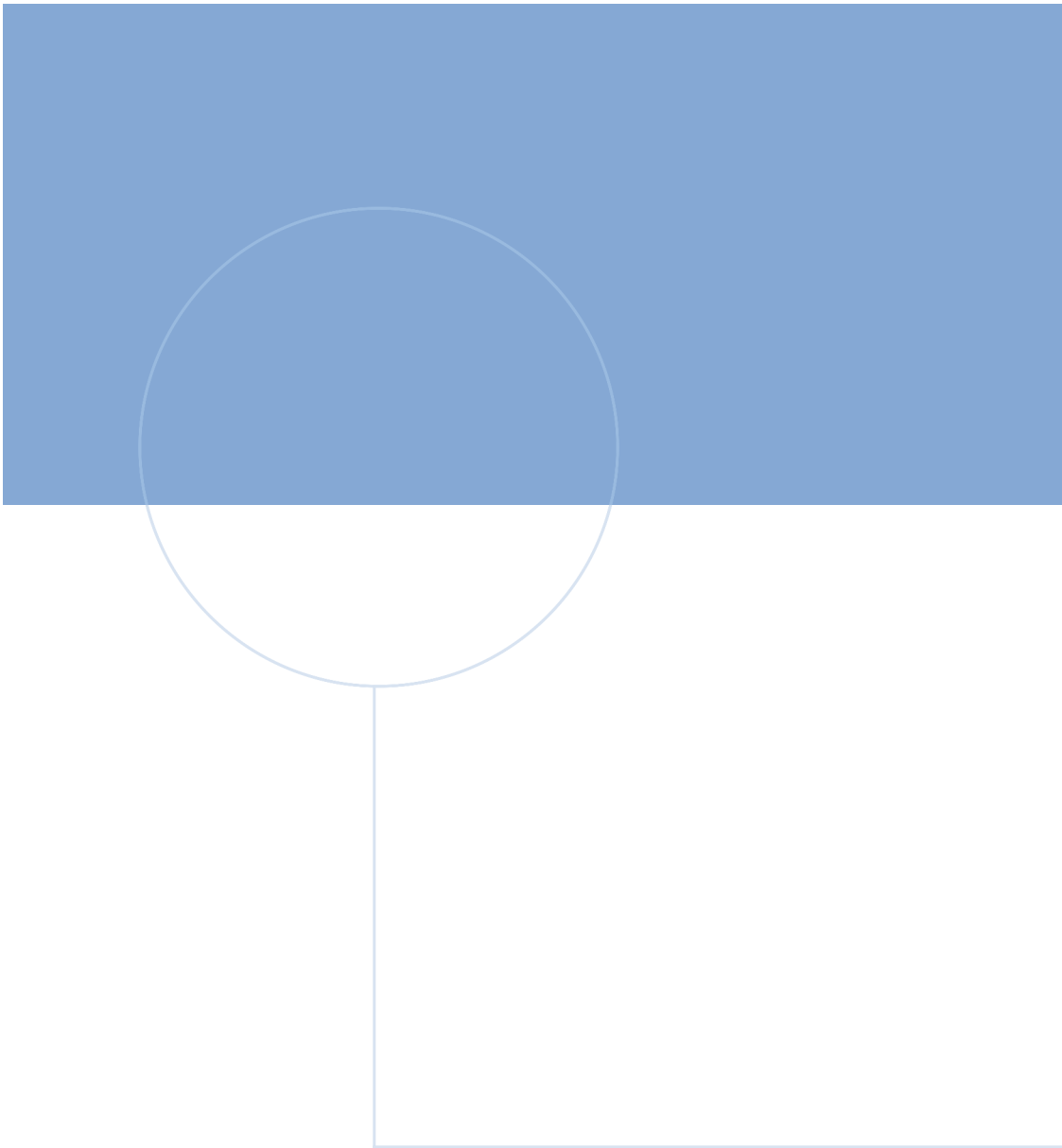
Table C.3: 3D Session Distance Results

Session	Inital Position		Position		Standard deviation		Initial Error	Error
	x	y	x	y	x	y		
3D-1	1451.7077	113.8320	61.8383	-1360.4627	283.4834	15384.1817	1421.3783	1441.3086
3D-2	-312.6559	67.3861	-1711.5562	31.3696	76040.4514	2033.3253	328.8014	1728.1652
3D-3	1361.8424	110.9225	58.9040	-1264.1817	282.1175	14992.6437	1331.4697	1344.9870
3D-4	-275.8727	68.4875	-1267.6131	40.4456	797470.9325	23191.9246	292.0020	1284.1382
3D-5	1594.6308	118.6082	67.0169	-1487.2595	335.0340	19790.3712	1564.3748	1568.1896
3D-6	-385.7945	65.1714	-1601.5458	28.2627	535680.0485	16455.2707	401.9731	1618.2830
3D-7	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3D-8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3D-9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3D-10	30.8674	91.1527	30.3831	83.9433	5.8837	22.3023	10.6437	3.4498
3D-11	30.6809	84.5866	30.4620	81.9783	5.7890	25.0871	4.0768	1.4908
3D-12	30.9240	89.1328	30.2913	86.0170	4.7402	28.4393	8.6252	5.5237
3D-13	77.0151	124.8151	75.6071	91.9267	33.3184	160.2919	43.5351	10.7336
3D-14	77.1083	124.6351	76.0405	90.5013	20.4243	166.9263	43.3553	9.2680
3D-15	77.1487	131.3188	75.9361	88.7767	30.3046	215.2045	50.0391	7.5677
3D-16	30.3511	63.9047	30.9665	74.1783	5.1028	10.3515	16.6094	6.3365
3D-17	30.7706	67.2732	27.6373	73.4977	4.8896	7.9755	13.2369	7.6599
3D-18	30.9243	66.0102	30.5705	74.3178	4.6904	8.2244	14.5013	6.1940
3D-19	33.7897	72.2440	34.8090	69.8757	7.3521	11.9211	8.8176	11.3933
3D-20	33.0251	67.5737	34.7317	68.4372	6.6535	8.6039	13.1400	12.7218
3D-21	33.0224	70.6159	33.5307	71.8756	5.9825	10.1571	10.1585	9.0803
3D-22	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3D-23	73.8364	64.4558	71.4974	70.2537	8.5160	11.6597	9.5116	4.0951
3D-24	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3D-25	72.8268	66.4481	71.3163	71.0684	8.9398	11.4242	7.5144	3.4759
3D-26	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3D-27	71.1415	70.2039	71.3214	70.9837	12.0769	14.2186	4.3037	3.5436
3D-28	74.4292	66.9541	76.2298	69.3118	19.9442	34.0866	7.0929	5.5075
3D-29	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3D-30	75.4152	65.5753	77.8013	66.2934	9.6226	15.1128	8.6476	8.9021
3D-31	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3D-32	73.8255	67.7810	76.8900	67.9446	24.7456	23.4607	6.1948	7.0173
3D-33	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3D-34	61.0391	67.7000	61.6907	70.3837	7.6464	22.3061	6.3999	3.6450
3D-35	58.3571	67.9412	55.8757	66.1364	9.2786	23.4113	7.1770	10.0960
3D-36	60.2387	68.3600	58.4753	67.4997	8.3144	9.3135	5.9685	7.4931
3D-37	63.5802	56.2957	64.2198	67.9227	13.5259	38.0747	17.7465	6.3884
3D-38	63.4989	54.3877	63.4407	68.7764	8.8247	25.8257	19.6439	5.3546
3D-39	63.6272	54.0103	64.5097	66.9841	15.2013	37.0239	20.0292	7.3706

Table C.4: 3D Session Adjustment Results

Session	Position		Standard Deviation		Error
	x	y	x	y	
3D-1	61.9170	-1340.4860	230.1649	15279.4461	1421.3384
3D-2	-2289.1823	14.3531	100957.4549	2777.4219	2306.0391
3D-3	58.9206	-1240.5525	196.0003	13439.1414	1321.3635
3D-4	-1400.6233	38.1866	83748.7354	2344.2387	1417.1606
3D-5	66.5944	-1472.3812	291.8858	18277.6586	1553.3055
3D-6	-4257.0470	-44.4202	218845.4021	6263.2968	4274.7762
3D-7	NaN	NaN	NaN	NaN	NaN
3D-8	NaN	NaN	NaN	NaN	NaN
3D-9	NaN	NaN	NaN	NaN	NaN
3D-10	30.6496	84.2256	6.7985	20.8755	3.7163
3D-11	30.4023	82.0916	5.7049	24.7313	1.6132
3D-12	30.6036	83.3599	6.3755	22.5032	2.8523
3D-13	89.0391	88.5151	47.3420	40.2866	14.0716
3D-14	80.1621	86.3188	28.9967	42.4570	5.9648
3D-15	79.5894	98.7701	11.0634	69.5779	17.6852
3D-16	30.4353	73.2146	5.9185	7.8945	7.3010
3D-17	30.4959	73.7110	5.7686	7.9513	6.8027
3D-18	30.9716	73.5380	5.7326	7.9558	6.9765
3D-19	33.0751	72.3832	5.2745	8.5882	8.4612
3D-20	32.6951	70.8353	5.5166	8.1934	9.8742
3D-21	32.5147	72.0985	5.3251	8.4410	8.6008
3D-22	NaN	NaN	NaN	NaN	NaN
3D-23	71.6370	71.4492	8.7005	15.3744	2.9813
3D-24	NaN	NaN	NaN	NaN	NaN
3D-25	71.4755	72.0447	8.7283	14.5616	2.6105
3D-26	NaN	NaN	NaN	NaN	NaN
3D-27	70.9878	70.6777	8.6598	13.3224	3.9838
3D-28	74.7287	69.9661	12.4590	20.4791	4.2460
3D-29	NaN	NaN	NaN	NaN	NaN
3D-30	75.5760	68.7271	13.2249	18.6216	5.7134
3D-31	NaN	NaN	NaN	NaN	NaN
3D-32	74.2999	70.8128	12.0950	21.7956	3.3050
3D-33	NaN	NaN	NaN	NaN	NaN
3D-34	60.8483	70.0167	6.9963	10.4977	4.2034
3D-35	58.8141	68.6066	7.2137	9.2884	6.3704
3D-36	60.1626	69.8133	7.0316	10.3399	4.6559
3D-37	62.8746	68.1051	8.6133	20.0308	5.9212
3D-38	62.9784	69.0362	8.3654	24.1517	5.0115
3D-39	63.6253	70.9347	8.2649	41.5669	3.3630

Table C.5: Un-Weighted Position Results



NTNU

Norwegian University of
Science and Technology