uc3m | Universidad Carlos III de Madrid

TESIS DOCTORAL

Opportunistic Timing Signals for Pervasive Mobile Localization

Autor:
Aymen Fakhreddine

Director:
Domenico Giustiniano

PROGRAMA DE DOCTORADO EN INGENIERÍA TELEMÁTICA

Leganés (Madrid), mayo de 2018

Universidad Carlos III de Madrid

PH.D. THESIS

Opportunistic Timing Signals for Pervasive Mobile Localization

Author:
Aymen Fakhreddine

Director:
Domenico Giustiniano

PHD PROGRAM IN TELEMATIC ENGINEERING

Leganés (Madrid), May, 2018

Opportunistic Timing Signals for Pervasive Mobile Localization

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Prepared by
Aymen Fakhreddine, IMDEA Networks Institute & University Carlos III of Madrid

Under the advice of
Domenico Giustiniano, IMDEA Networks Institute

Date: May, 2018

Web/contact: [aymen.fakhreddine@imdea.org]

This work has been supported by IMDEA Networks Institute.

uc3m | Universidad Carlos III de Madrid

TESIS DOCTORAL

Opportunistic Timing Signals for Pervasive Mobile Localization

***Autor*: Aymen Fakhreddine**

Director: Domenico Giustiniano

Firma del tribunal calificador:

Firma

Presidente:

Vocal:

Secretario:

Calificación:

Leganés, de de

Acknowledgements

First of all, I would like to express my deepest gratitude to my PhD director Domenico Giustiniano who guided me all along my PhD journey. He was always generous by continuously providing constructive feedback, careful reviews of my papers and creative and challenging research ideas. Thank you for being my mentor, for your encouragements and patience with me, and for everything I learned from you.

Very special thanks go to Vincent Lenders for his insights and pertinent comments and feedbacks that were always of a tremendous help. He always had a special perspective on the topics we worked on that made the papers we co-authored highly distinguished. Thanks also for organizing the CyberAlpRetreat event to which I was invited 3 years in a row. It was always a propitious opportunity to learn and grow as a researcher.

I am deeply grateful to Nils Ole Tippenhauer who hosted me at the Singapore University of Technology & Design. Apart from his immense hospitality, he was always available to provide me with precious help, assistance and advice. I spent exceptional time whether at the professional or personal level. Thank you Nils, it was an honor collaborating with you.

I would like to thank all people from IMDEA Networks Institute. You have made my daily life in the office particularly enjoyable. Very special thanks go to Roberto Calvo and Maurizio Rea, they are not simply co-authors but also very good friends.

Last but not least, I want to express my deep gratitude to my parents who always ensured my well-being. You have given me confidence since I was born, I love you. I also thank my family for all their encouragements. You have always been my source of inspiration.

Abstract

The proliferation of handheld devices and the pressing need of location-based services call for precise and accurate ubiquitous geographic mobile positioning that can serve a vast set of devices. Despite the large investments and efforts in academic and industrial communities, a pin-point solution is however still far from reality. Mobile devices mainly rely on Global Navigation Satellite System (GNSS) to position themselves. GNSS systems are known to perform poorly in dense urban areas and indoor environments, where the visibility of GNSS satellites is reduced drastically. In order to ensure interoperability between the technologies used indoor and outdoor, a pervasive positioning system should still rely on GNSS, yet complemented with technologies that can guarantee reliable radio signals in indoor scenarios. The key fact that we exploit is that GNSS signals are made of data with timing information. We then investigate solutions where opportunistic timing signals can be extracted out of terrestrial technologies. These signals can then be used as additional inputs of the multi-lateration problem. Thus, we design and investigate a hybrid system that combines range measurements from the Global Positioning System (GPS), the world's most utilized GNSS system, and terrestrial technologies; the most suitable one to consider in our investigation is WiFi, thanks to its large deployment in indoor areas. In this context, we first start investigating standalone WiFi Time-of-flight (ToF)-based localization. Time-of-flight echo techniques have been recently suggested for ranging mobile devices over WiFi radios. However, these techniques have yielded only moderate accuracy in indoor environments because WiFi ToF measurements suffer from extensive device-related noise which makes it challenging to differentiate between direct path from non-direct path signal components when estimating the ranges. Existing multipath mitigation techniques tend to fail at identifying the direct path when the device-related Gaussian noise is in the same order of magnitude, or larger than the multipath noise. In order to address this challenge, we propose a new method for filtering ranging measurements that is better suited for the inherent large noise as found in WiFi radios. Our technique combines statistical learning and robust statistics in a single filter. The filter is lightweight in the sense that it does not require specialized hardware, the intervention of the user, or cumbersome on-site manual calibration. This makes the method we propose as the first contribution of the present work particularly suitable for indoor localization in large-scale deployments using existing legacy WiFi infrastructures. We evaluate our technique for indoor mobile tracking scenarios in multipath environments, and, through extensive evaluations across four different testbeds covering areas up to $1000\,\mathrm{m}^2$,

the filter is able to achieve a median ranging error between 1.7 and 2.4 meters.

The next step we envisioned towards preparing theoretical and practical basis for the aforementioned hybrid positioning system is a deep inspection and investigation of WiFi and GPS ToF ranges, and initial foundations of single-technology self-localization. Self-localization systems based on the Time-of-Flight of radio signals are highly susceptible to noise and their performance therefore heavily rely on the design and parametrization of robust algorithms. We study the noise sources of GPS and WiFi ToF ranging techniques and compare the performance of different self-positioning algorithms at a mobile node using those ranges. Our results show that the localization error varies greatly depending on the ranging technology, algorithm selection, and appropriate tuning of the algorithms. We characterize the localization error using real-world measurements and different parameter settings to provide guidance for the design of robust location estimators in realistic settings.

These tools and foundations are necessary to tackle the problem of hybrid positioning system providing high localization capabilities across indoor and outdoor environments. In this context, the lack of a single positioning system that is able the fulfill the specific requirements of diverse indoor and outdoor applications settings has led the development of a multitude of localization technologies. Existing mobile devices such as smartphones therefore commonly rely on a multi-RAT (Radio Access Technology) architecture to provide pervasive location information in various environmental contexts as the user is moving. Yet, existing multi-RAT architectures consider the different localization technologies as monolithic entities and choose the final navigation position from the RAT that is foreseen to provide the highest accuracy in the particular context. In contrast, we propose in this work to fuse timing range (Time-of-Flight) measurements of diverse radio technologies in order to circumvent the limitations of the individual radio access technologies and improve the overall localization accuracy in different contexts. We introduce an Extended Kalman filter, modeling the unique noise sources of each ranging technology. As a rich set of multiple ranges can be available across different RATs, the intelligent selection of the subset of ranges with accurate timing information is critical to achieve the best positioning accuracy. We introduce a novel geometrical-statistical approach to best fuse the set of timing ranging measurements. We also address practical problems of the design space, such as removal of WiFi chipset and environmental calibration to make the positioning system as autonomous as possible. Experimental results show that our solution considerably outperforms the use of monolithic technologies and methods based on classical fault detection and identification typically applied in standalone GPS technology.

All the contributions and research questions described previously in localization and positioning related topics suppose full knowledge of the anchors positions. In the last part of this work, we study the problem of deriving proximity metrics without any prior knowledge of the positions of the WiFi access points based on WiFi fingerprints, that is, tuples of WiFi Access Points (AP) and respective received signal strength indicator (RSSI) values. Applications that benefit from proximity metrics are movement estimation of a single node over time, WiFi fingerprint matching for localization systems and attacks on privacy. Using a large-scale, real-world WiFi fingerprint data set consisting of 200,000 fingerprints resulting from a large deployment of wearable WiFi sensors, we show that metrics from related work perform poorly on real-world data. We analyze the cause for this poor performance, and show that imperfect observations of APs with commodity WiFi clients in the neighborhood are the root cause. We then propose improved metrics to provide such proximity estimates, without requiring knowledge of location for the observed AP. We address the challenge of imperfect observations of APs in the design of these improved metrics. Our metrics allow to derive a relative distance estimate based on two observed WiFi fingerprints. We demonstrate that their performance is superior to the related work metrics.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ACK	**ACK**nowledgement
AGPS	**A**ssisted **G**lobal **P**ositioning **S**ystem
AoA	**A**ngle **o**f **A**rrival
AP	**A**ccess **P**oint
API	**A**pplication **P**rogramming **I**nterface
CAESAR	**CA**rri**E**r **S**ense-b**A**sed **R**anging
COTS	**C**ommercial **O**ff-**T**he-**S**helf
CSMA/CA	**C**arrier-**S**ense **M**ultiple **A**ccess with **C**ollision **A**voidance
DLL	**D**elay-**L**ocked **L**oop
ECDF	**E**mpirical **C**umulative **D**istribution **F**unction
ECEF	**E**arth-**C**entered, **E**arth-**F**ixed
EKF	**E**xtended **K**alman **F**ilter
EM	**E**xpectation **M**aximization
ESS	**E**xtended **S**ervice **S**et
EWMA	**E**xponentially **W**eighted **M**oving **A**verage
FFT	**F**ast **F**ourier **T**ransform
GMM	**G**aussian **M**ixture **M**odel
GNSS	**G**lobal **N**avigation **S**atellite **S**ystem
GPS	**G**lobal **P**ositioning **S**ystem
HDOP	**H**orizontal **D**ilution **O**f **P**recision
KF	**K**alman **F**ilter
LBS	**L**ocation-**B**ased **S**ervice
LLA	**L**atitude, **L**ongitude and **A**ltitude
LOS	**L**ine-**O**f-**S**ight
LS	**L**east **S**quares
LTE	**L**ong-**T**erm **E**volution
MAC	**M**edia **A**ccess **C**ontrol
MUSIC	**MU**ltiple **SI**gnal **C**lassification
NIC	**N**etwork **I**nterface **C**ontroller
NLOS	**Non**-**L**ine-**O**f-**S**ight

NSE	**N**ational **S**cience **E**xperiment
PHY	**PHY**sical layer
PRN	**P**seudo-**R**andom **N**umber
QQ	**Q**uantile **Q**uantile
RAIM	**R**eceiver **A**utonomous **I**ntegrity **M**onitoring
RAT	**R**adio **A**ccess **T**echnology
RF	**R**adio **F**requency
RFID	**R**adio-**F**requency **ID**entification
RINEX	**R**eceiver **IN**dependent **EX**change
RMSE	**R**oot-**M**ean-**S**quare **E**rror
RSS	**R**eceived **S**ignal **S**trength
RSSI	**R**eceived **S**ignal **S**trength **I**ndicator
SAIL	**S**ingle **A**ccess point-based **I**ndoor **L**ocalization
SHM	**SH**ared **M**emory
SIFS	**S**hort **I**nter**F**rame **S**pace
SNR	**S**ignal-to-**N**oise **R**atio
SSID	**S**ervice **S**et **ID**
SV	**S**pace **V**ehicle
ToA	**T**ime-**o**f-**A**rrival
ToF	**T**ime-**o**f-**F**light
TTFF	**T**ime **T**o **F**irst **F**ix
UDP	**U**ser **D**atagram **P**rotocol
WLAN	**W**ireless **L**ocal **A**rea **N**etwork

Chapter 1

Introduction

The aim of this thesis is the design and evaluation of a full-fledged system architecture for pervasive mobile positioning. This problem is timely since the rapid proliferation of smartphones, tablets and more recently wearable devices is driving the market to create innovative services. In particular, the geographical location is at the core of a number of Location Based Services (LBS) on mobile devices, in a market expected to reach 40.99 billion US dollars by 2022 while it only accounted for 5.22 billion US dollars in 2016 [2].

Yet, a seamless positioning system is far from reality: the ossification of scientific approaches for indoor localization is proved by the fact that in the last decade no significant progresses have been made in the commercial market out of a vast research literature, where services provide similar accuracy as years ago and improvements are mainly in terms of coverage and mapping of indoor buildings [67], and ”new” solutions such as iBeacon still rely on the cell identifier messages broadcasted by nodes such as iPhones [1]. The above shortcomings are caused by the lack of pervasive support of the output of a navigation system (e.g. the position) at the quality (for instance: positioning accuracy) requested by the mobile services. We motivate that through a simple but concrete problem, which prevents from realizing a seamless experience. (P) A user walks in the city center until she enters in some narrow roads with three-floors buildings. GPS originally worked fine, and then started displaying a wrong position, until no location estimate is available. Mobile sensors are also not helpful, since they continuously require accurate landmarks to compensate for dead reckoning.

There are plenty of terrestrial radio signals in range in the aforementioned areas. Why are they not helpful when needed? In simple terms, problem (P) is caused by the general low accuracy of satellite navigation systems in absence of clear view of the sky, and the inability of current systems to use the right data from nearby space. Existing scientific approaches based on WLAN networks can only deliver median accuracy in the order of a few meters or being environmental-specific calibration intensive [8, 15, 80]. More demanding metrics, such as the 80-percentile of precision, often provide unacceptable results.

As a potential killer use-case, we imagine to be in the gray zone, where signals from one positioning technology do not suffice to locate the device. Supposing we can receive signals from two GPS satellites and from 2 WLAN APs, 3D-multilateration cannot be computed with only two independent signals from one technology. In contrast, it can be computed with four independent signals from two technologies (GPS and WLAN). As a result, the mobile could now position itself. However, current devices do not allow such an integration since the different technologies work as monolithic blocks. We address this fundamental problem in the thesis. Therefore, within the timeline of the present thesis, we investigate and demonstrate the feasibility of a fine-grained and seamless mobile positioning with accuracy and coverage not possible nowadays. We provide results applicable to real-world scenarios and, therefore, of interest for academia and industry. Hereby, manufacturers of mobile devices (smartphones, tablets, etc,...), wearable cognitive assistance (glasses, watches, etc,...) and network infrastructure (APs, controllers, etc,...), are envisioned first adopters. Users and LBS will all benefit from the new concrete vision and tangible results that this thesis brings.

As of today, satellite navigation and data network communication have operated according to isolated structures. Satellite navigation systems process the incoming satellite signals and computes the PVT (Position, Velocity, Time) data. Conversely, technologies such as Wireless Local Area Network (WLAN) and cellular networks are natively designed to address the communication problem, and, just more recently, they provide support for simple navigation. However, their focus remain on communication. In the realm of navigation, Global Positioning System (GPS) is the ”de-facto” standard technology, while it is well-known to fail indoor. Interest for the indoor location ecosystem is mainly spawn by the need of users to navigate indoor, as well as social applications, emergency response and augmented reality. An arising area of interest that requires precise indoor location is physical analytics, that aims to understand and better plan the physical space [56]. Given the additional benefit of the densification of APs deployment and the large availability of WLAN chipsets in most today’s mobile devices, the result is that 802.11 WLAN is currently the driving assisting technology for location services, also beating other short range communication technologies like infrared [73], RFID [36], ultrasonic [63] and ultra-wideband [27]. In the commercial sector, big industrial players, such as Google, have introduced indoor navigation services based on 802.11 WLAN and indoor maps. There has been limited attention in the literature to introduce an architecture to integrate data and signals of terrestrial communication and satellite navigation signals at a high granularity [69]. The most successful approach so far is Assisted-GPS (AGPS), that greatly helps to reduce the time-to-first fix (TTFF) by sending ephemeris and almanac data through communication networks. These approaches do not help to neither increase the coverage nor the accuracy of the navigation service, and do not integrate signals but rather data [58]. Standards of cellular networks such as LTE also provide support for timing information, but there has been little exploitation so far for commercial services.

1.1. Publications that have led to this Thesis

The ideas and investigations briefly described above resulted in the following refereed publications [22–24, 65]. All papers are already published. [65] is published in *IEEE/ACM Transactions on Networking*, a **Q1 journal**. Moreover, the author presented [22] at *IEEE WiOpt 2016*, [23] at *ACM/IEEE IPSN 2018* and [24] at *European Wireless 2018*. *IEEE WiOpt* and *ACM/IEEE IPSN* conferences are respectively rated **B** and **A*** according to *CORE 2018* conferences ranking. Finally, [13] is a contribution that emanated from a collaboration with Roberto Calvo, PhD student at the same university. It is published in *IEEE INFOCOM 2017* (*CORE 2018* **A***) but is not considered as part of the present PhD thesis report submitted for evaluation.

- [24] A.Fakhreddine, N.O.Tippenhauer, D.Giustiniano, ”Design and Large-Scale Evaluation of WiFi Proximity Metrics”, *EUROPEAN WIRELESS 2018*, Catania, Italy, May 2018.

- [23] A.Fakhreddine, D.Giustiniano, V.Lenders, ”Data Fusion for Hybrid and Autonomous Time-of-Flight Positioning”, *IEEE/ACM IPSN 2018*, Porto, Portugal, April 2018.

- [65] M.Rea, A.Fakhreddine, D.Giustiniano, V.Lenders, ”Filtering Noisy 802.11 Time-of-Flight Ranging Measurements with Commoditized WiFi Radios”, *IEEE/ACM Transactions on Networking 2017*.

- [22] A.Fakhreddine, D.Giustiniano, V.Lenders, ”Evaluation of Self-Positioning Algorithms for Time-of-Flight based Localization”, *WiOpt 2016*, Tempe, Arizona, USA, May 2016.

- [13] R.Calvo-Palomino, D.Giustiniano, V.Lenders, A.Fakhreddine, ”Crowdsourcing Spectrum Data Decoding”, *IEEE INFOCOM 2017*, Atlanta, GA, USA, 1-4 May 2017.

1.2. Main Contributions and Organization of the Thesis

We highlight in this section the principal findings of this thesis and how the latter is structured.

Chapter 2 discusses the notions and research questions we attempt to tackle in this thesis, and related work from the literature on the general topic regarding pervasive and indoor localization, with a special focus on WiFi ToF-based ranging and technology fusion for positioning.

In Chapter 3 we provide a thorough analysis and experimental results related to WiFi Time-of-Flight ranging with Commercial-Off-The-Shelf (COTS) Wifi radios. Apart from the last Section of Chapter 3 (Section 3.8), the content of this Chapter is essentially published in [65], a work co-authored with the PhD student Maurizio Rea. We point out that the contributions of the author of the present thesis to the experiments shown in Section 3.3.3, Section 3.3.4 and Section 3.3.5 are very limited, nevertheless, we show them in the thesis for completeness and a smooth flow of ideas and contributions. The author rather worked on the modeling of the non-Gaussian noise affecting the ToF measurements, contributed to the design and evaluation of the adaptive filter for ToF shortest path estimator (Section 3.7) and proposed the environment training-free shortest path estimator (Section 3.8). After introducing principles of the WiFi echo technique for ToF computation, an exhaustive investigation of the noises affecting the ToF measurements is presented in Section 3.3. This is a prerequisite for a design of a ToF shortest path estimator. Two methods are proposed, an adaptive filter based on environment training described in Section 3.7, and an environment calibration-free shortest path estimator built on a robust Gaussian Mixture Model (GMM) fit.

Chapter 4 poses the problem of hybrid positioning. As the considered technologies for this hybrid positioning system are GPS and WiFi ToF (see Chapter 3), an overview of the well-known GPS ranging is given in this Chapter in order to make this thesis book self-contained. Before digging into the multi-technology multilateration problematic, we first inspect the Extended Kalman filter (EKF) in the context of single-technology positioning and compare it to the widely used Least-Squares approach. A study of the EKF parameters' tuning is also presented in Section 4.5 as this step is crucial to optimize the EKF and later on tune it properly in the context of fusing heterogeneous ranges. This is exhaustively analyzed in Section 4.6. Another relevant problematic we examine here and illustrate in Section 4.7.1 is how to estimate the bias introduced by the WiFi AP while using the ToF echo technique, discussed in the previous Chapter. We propose to alleviate the need of any knowledge concerning this bias which is chipset dependent and estimate it taking benefit of GPS ranges, we also show a fingerprinting approach to compute this chipset bias in the context of pure WiFi positioning. Finally, in the scenario in which an abundant number of anchors (GPS SVs and WiFi APs) is available, we come up with a novel Statistical-Geometrical method (Section 4.9) to infer the optimal set of anchors to consider for the position fix and thus remove the unreliable ones that negatively affect the positioning accuracy.

Chapter 5 depicts the design and large-scale evaluation of WiFi-proximity metrics in a context in which the positions of the anchors are unknown. After illustrating the real and artificial datasets used for this study, we show how related work metrics are inaccurate in the considered context. Then we show that the probabilistic observation of APs is the main cause of this inaccuracy. Thus, we address this issue in the design of improved proximity metrics in Section 5.4.3 and demonstrate their accuracy and robustness.

Finally, in Chapter 6, we wrap-up the main ideas and concepts studied in this thesis, draw relevant conclusions before giving ideas of future work that could potentially extend this work.

Chapter 2

Background & Related Work

2.1. Indoor Localization

The indoor localization literature is vast, including techniques using signal strength [7, 15, 32, 50, 80], the angle of arrival [30, 66, 78] or combining WiFi signals with inertial sensors as found in smartphones [64]. In this section, we aim to give an brief overview on different indoor localization techniques in addition to some general Non-Line-Of-Sight (NLOS) mitigation based solutions.

Common proposals to combat the multipath problem are for example the use of ultrawideband signals [18,27,46] or frequency diversity [45,62]. However, these approaches require specialized hardware or software-defined radios which increase costs and hinders localization at larger scales. ToneTrack [79] tries to overcome the problem of limited bandwidth, inherent to WiFi time-based localizations. It combines channels to form virtual larger bandwidths without increasing the radio's sampling rate, taking benefit of frequency hopping, to increase the resolution of Time of Arrival (ToA) profiles. SpotFi [44] uses APs equipped with 3 antennas and commodity WiFi chipsets. It jointly estimates the Angle of Arrival (AoA) and ToF pairs of each path using the channel state information, and estimates the likelihood that these pairs correspond to the direct path between the AP and the target. [44] does not use ToF to compute the ranges, so its utility is limited to the likelihood estimation unlike our solutions presented in Sections 3.7 and 3.8 where the ranges are estimated from filtered ToF measurements. The need for intervention of the user may be unwanted in setups where the objective of positioning the user is to understand and better plan the physical space [56]. A solution that can independently run on the infrastructure side has also the advantage of disentangling the core of the positioning tracking from the mobile application where additional services and higher accuracy could be provided only if the user is interested to install the application. Other solutions try to identify and mitigate the NLOS effects like [77] that uses WiFi RSS measurements to do so. The proposed solution combines a machine learning technique to first extract typical features from the training data collected during extensive indoor measurement campaigns and estimate the ranges using a regression model, and an identification approach based on hypothesis testing. The approach still needs a training phase that requires an offline classification or calibration. Still in RSS based ranging, [72] proposes to use a GMM model to filter corrupted range estimations caused by NLOS radio propagation by modeling distributions of LOS and NLOS sets of estimates. To mitigate the NLOS effect based on ToA measurements, in [37] the authors propose to use in a multiple model framework two filters in parallel, an Extended Kalman Filter (EKF) that performs well in LOS environments and a robustified EKF based on a regression that uses a redescending score function to deal with harsh NLOS environments.

In the following, we classify different indoor localization approaches in key categories. Solutions requiring specialized hardware, Angle-of-Arrival techniques, RSS-based WiFi localization and we finally dedicate the next section to WiFi ToF techniques.

Solutions requiring specialized hardware. The earliest indoor localization schemes were based on specialized hardware. Cricket [63] used ultrasound devices and the Active Badge [73] system relied on infrared equipment. RFID-based [36] or ultra-wideband systems [27] have also been proposed. While dedicated systems with specialized hardware may achieve higher accuracy than techniques that rely on signals from existing infrastructures such as WLAN or GSM, their application is limited to dedicated deployments. 802.11 WLAN is the driving assisting technology for location services, also beating other short range communication technologies like infrared [73], RFID [36], ultrasonic [63] and ultra-wideband [27].

RSS-based WiFi localization techniques. The research community has been striving to find a pin-point positioning solution for WiFi based on the radio-frequency signal strength. A drawback of using the RSSI is that the signal attenuation in indoor environments becomes not only depending on the distance but also highly dependent on the material of the obstacles between radio devices like walls, doors, etc. As a consequence, RSSI-based indoor localization systems require extensive site-specific calibration, in order to match RSSI measurements against the training data. Several matching techniques have been proposed, either deterministically such as k-nearest neighbor [7], or probabilistically such as Bayesian inference [80], or using side-channel information like inertial sensors and maps [39, 51, 64]. Recent effort to remove the offline radio calibration is leading to models which usually result in lower accuracy [15, 32, 50]. Studies conducted by [81] have shown that commercial products are less accurate (median error of 74 m) and may fail to provide the published accuracy specifications. Over the last years, many WiFi-based localization systems have been proposed [50, 64, 80]. The vast majority of these systems rely on the Received Signal Strength Indicator (RSSI) to estimate the position.

Angle of arrival localization techniques. Recent approaches in 802.11 also exploit the Angle of Arrival (AoA), which requires advancements in the hardware. ArrayTrack uses sophisticated rectangular array of 16 antennas [78]. CUPID [66] works on COTS APs and it reduces the effect of multipath with respect to signal-strength measurements. However it is highly susceptible to indoor shadowing, which results in distance errors of more than 10 m. Differently from our system, it further needs inputs from the inertial sensors of the smartphones. [30] is a work that addresses some of the practical challenges to apply AoA in commodity hardware, it shows that 802.11 NICs can be calibrated and synchronized to a tolerance of 15 degrees median phase error using three and five antennas. AWL [14] uses a spatial aliasing scheme to improve the AoA estimation accuracy, it achieves a sub-meter accuracy using a single AP equipped with 6 antennas.

2.2. ToF-based WiFi localization

ToF-based localization holds the promise of WLAN-based localization without environmental calibration requirements. ToF Echo techniques based on packet exchanges in WiFi networks were first proposed in [49, 54] and refined in [16, 29, 31, 34]. However, unlike our work, none of these approaches address the effect of non-Gaussian noise such as in multipath-rich environments. ToF measurements are subject to severe noise and have thus long been considered impracticable for WLAN localization. Back then, Li et al. [49] suggested in their work that the accuracy of spacing between a DATA and ACK frame that are defined to be separated by a SIFS time is up to 2 μs, equal to 600 m of error, and too high for indoor localization. Later on, echo techniques have however been proposed by [54] to resolve the absence (or low precision) of clock synchronization in Wi-Fi chipsets. Because of hardware limitations and limited access to low-level functionalities, ToF measurements conducted by [16, 31, 34] were subject to variable jitter and significant post-processing. In [26], the authors introduced directional Yagi antennas to eliminate the effect of multipath and other noise sources from WiFi echo techniques. They achieved a positioning accuracy of less than 5 m in 8 from 10 positions. In contrast, the filter and shortest path estimator we introduce in Chapter 3 work with omnidirectional antennas in environments with multipath.

Our approach has the additional advantage of reducing the time estimation uncertainty for weak signal-to-noise ratios transmissions. More recently, CAESAR [29] has shown a methodology to extract ToF measurement using driver-level operations. A direct comparison to [29] as presented in this work (Section 3.7.3) shows that the error with our statistical filter can be reduced in indoor environments by a factor of more than two compared to classical estimators that do not compensate for the bias of the multipath. [26] introduced directional Yagi antennas to eliminate the effect of multipath and other noise sources from WiFi echo techniques. They achieved a positioning accuracy of less than 5 m in 8 positions over 10. In contrast, our system works with single omnidirectional antennas in environments with multipath. SAIL [53] is a ToF system using WiFi that has been designed for localization in multipath environments. However, SAIL requires inputs from the inertial sensors in the smartphone. SAIL achieved median error of $\approx$ 1 m and 80-percentile error of $\approx$ 5 m, which is comparable to our filter, at the drawback of requesting the collaboration from the mobile user through the installation of a dedicated application on smartphones. The need for intervention of the user can be unwanted in setups where the objective of positioning the user is to understand and better plan the physical space [56]. A solution that can independently run on the infrastructure side has also the advantage of disentangling the core of the positioning tracking from the mobile application where additional services and higher accuracy could be provided only if the user is interested to install the application.

2.3. Fusion of different technologies

As of today, satellite navigation and data network communication have operated according to isolated structures. Satellite navigation systems process the incoming satellite signals and computes the position Conversely, technologies such as WiFi and cellular networks are natively designed to address the communication problem, and, just more recently, they provide support for simple navigation. However, their focus remain on communication. In the realm of navigation, Global Positioning System (GPS) is the 'de-facto' standard technology, while it is well-known to fail indoor. Interest for the indoor location ecosystem is mainly spawn by the need of users to navigate indoor, as well as social applications, emergency response and augmented reality. The types of assistance from radio-terrestrial communication technologies can be segmented in either through assistance and measurements from the infrastructure or from the user. Given the additional benefit of the densification of APs deployment and the large availability of WiFi chipsets in most today's mobile devices, the result is that 802.11 WiFi is currently the driving assisting technology for location services.

Techniques based of ranges include signal strength [7, 15, 32, 50, 80], angle of arrival [30, 66, 78] and time-of-flight (ToF) [29, 53, 79]. There has been limited attention in the literature to introduce an architecture to integrate data and signals of terrestrial communication and satellite navigation signals at a high granularity [69]. The most successful approach so far is Assisted-GPS (AGPS), that greatly helps to reduce the time-to-first fix (TTFF) by sending ephemeris and almanac data through communication networks. These approaches do not help to neither increase the coverage nor the accuracy of the navigation service, and do not integrate signals but rather data [58]. To this end, solutions such as real-time differential GPS are necessary, which can minimize some of the systematic errors. These methods are complementary to our work as they correct any bias at one location by measuring the bias at known locations and using a network of additional satellites and ground-based reference stations [55]. This highly reduces the errors affecting the GPS pseudoranges, however, they remain inefficient in scenarios with reduced GPS visibility. There has also been very little attention to the problem of understanding how to merge timing measurements for ranges coming from different technologies. [25] studies the ambiguity in the position that may be caused by using both GPS and WiFi range measurements, yet it does not try to outperform standalone GPS's accuracy. Some works like [15] investigate complementarity between GPS and WiFi positioning, but not a ranging-level fusion. [15] proposes a localization algorithm that relies on GPS to infer the WiFi APs positions when localizing mobile devices using WiFi signal strength. Our system can be extended introducing similar functionalities. Combining GPS with inertial sensors is widely used in outdoor environments to enhance the GPS positioning accuracy [5].

Chapter 3

WiFi ToF-based Ranging with COTS Radios

3.1. Introduction

Localization using the Time-of-Flight (ToF) of RF signals is today the most popular technique to track moving objects in outdoor environments. The most prominent usage of ToF is the Global Positioning System (GPS) which exploits signal propagation times from different satellites to localize mobile devices on earth. Also, radar systems commonly rely on the propagation time of RF signals to localize aircrafts in the airspace. While the ToF technique has been widely adopted in outdoor environments, its application for indoor localization in WiFi environments has been relatively modest so far. When it comes to WiFi based localization, the research community has instead focused more intensively on different approaches such as the signal strength [7,15,32,50, 80], the angle of arrival [30,44,66,78] or combining WiFi signals with inertial sensors as found in smartphones [53, 64]. The problem with the latter approaches is that they either require extensive manual on-site calibration, the need for intervention of the user, or specialized hardware. These factors limit the deployment of these approaches at larger scale.

The challenge with ToF is that very precise signal propagation time estimates are required. At the speed of light, a measurement error of 1 μs already results in a distance estimation error of 300 meters which is intolerable for most indoor applications. In order to achieve meter-level localization accuracy, a precision in the order of a few nanoseconds is therefore needed. However, at this level the ToF is affected by various sources of noise [16, 29, 31, 34, 49, 54]. For example, the relatively small bandwidth of WiFi signals and the limited clock rate of commercial off-the-shelf (COTS) WiFi radio receivers hinder the exact determination of the time of arrival of a signal. In echo techniques, the target may further add significant jitter before acknowledging the reception of the echo. Specially indoors, the signal may additionally be obstructed and reflected over multiple paths which adds a positive bias to the estimated range.

In order to mitigate the impact of noise in ToF measurements, several techniques have been proposed in the literature. To combat the device-related Gaussian noise, unbiased estimators that rely on the mean or median of the collected ToF samples provide a good estimate of the true ToF [29]. To combat the multipath noise, biased estimators that aim at identifying the direct path such as the MUSIC algorithm [48, 79] can provide good results as well. Other approaches such as the expectation-maximization algorithm have been successfully applied to signal strength-based localization [32], and we could envision their application to combat the multipath noise of ToF measurements. However, we show in this work that, in WiFi ToF, the noise follows a Gaussian mixture model with each Gaussian component being in the same order of magnitude as the multipath bias, and we demonstrate that these estimators fail to provide an accurate estimate of the true ToF as we need for ranging.

In this chapter, we present two filters that are specially designed for estimating the range in WiFi indoor environments based on noisy ToF measurements. Our first filter relies on a combination of statistical learning techniques to train an environmental-specific linear regression model for the multipath bias and robust statistics for range estimation. This approach does not require any specialized hardware and relies exclusively on the ToF information that can be extracted from COTS WiFi radios. In addition, our environmental-specific linear regression model can be learned in-situ and thus does not require any manual calibration. Moreover, we challenged this result towards a new ranging technique (second filter) that does not require any environment calibration while keeping the accuracy at an acceptable level.

We have used our filter to estimate the ranges of WiFi radio devices to the APs in range in three different experimental testbeds. Our results show that our filtering technique is able to significantly reduce the median ranging error over classical estimators. We further compared our new environment-training-free ranging technique to the previous filtering technique we propose in the largest of the previous testbeds. Our main contributions in this chapter are the following:

- We present a firmware-based ranging architecture for round-trip time measurements running in the MAC processor of WiFi chipsets, and we quantify and compare the amount of noise that comes from the WiFi devices and the noise from the WiFi radio signal propagation over the wireless channel.

- We propose a filter based on statistical learning and robust statistics to estimate the distance range from a series of noisy ToF measurements. Our filter does not require any manual calibration and manages to estimate the distance to a remote WiFi device with median error between 1.7 and 2.4 meters.

- Finally, we propose a GMM fit based ranging method that does not require environment training while guaranteeing a very insignificant loss in accuracy with respect to the previous filter that calibrates the environment.

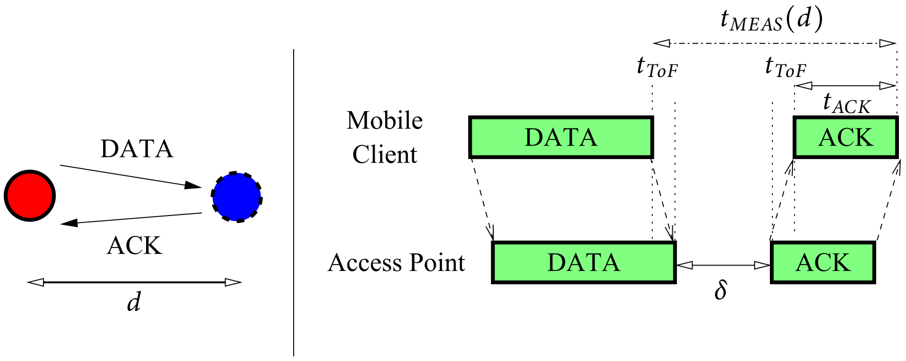


Figure 3.1: Principle of WiFi ToF echo technique

3.2. WiFi ToF Echo Technique

This section describes the principles of ToF ranging and the noise effects that arise when applying this technique to off-the-shelf WiFi devices.

While traditional ToF-based echo techniques as employed in radar systems rely on uncoded RF signals and their reflections, WiFi echo techniques use regular frames of communication [16, 29, 31, 34, 49, 54]. In WiFi communication, every DATA frame is acknowledged by the receiver with an ACK frame. Since the interframe time between the DATA and ACK frames is fixed by the 802.11 standard (the Short InterFrame Symbol, SIFS, time), the delay between DATA and ACK frames can be used to infer the distance between two nodes.

In reality, sources of time offset exist at the local (mobile client) and target (access point) stations and that affects the accuracy of the ranging measurement. Let us refer to Fig. 3.1. If d is the true distance between a local station and a target, the local station measures the time $t_{MEAS}(d)$ between a sent DATA frame and a received ACK frame:

$$t_{MEAS}(d) = 2 \cdot t^{ToF}(d) + t_{ACK} + \delta, \tag{3.1}$$

where $t^{ToF}(d)$ is the signal propagation time between the transmitter of the DATA frame and the target (channel reciprocity is assumed), t_{ACK} is the time needed to transmit the ACK, and δ is the measurement offset.

The offset δ is the sum of the offset due to the local station δ_L and the offset due to the target station δ_T:

$$\delta = \delta_L + \delta_T. \tag{3.2}$$

The local offset δ_L arises at the measuring device due to the local imprecision in the timing information extraction, and the target offset δ_T is equal to the SIFS time plus any device-dependent deviation from this number. The presence of multipath effect causes a positive offset to single measurements and increases δ_L and/or δ_T.

From eq. (3.1), we can then define the ToF measurement as

$$t^{ToF}(d) = \frac{t_{MEAS}(d) - t_{ACK} - \delta}{2}. \tag{3.3}$$

At distance d = 0, we then have

$$t^{ToF}(d=0) = \frac{t_{MEAS}(d=0) - t_{ACK} - \delta}{2} = 0. \tag{3.4}$$

And therefore eq. (3.3) can be rewritten as:

$$t^{ToF}(d) = \frac{t_{MEAS}(d) - t_{MEAS}(d=0)}{2}. \tag{3.5}$$

The distance from the measuring station to the target device can be computed for the generic sample[1] m as:

$$d_m = c \cdot t_m^{ToF}(d), \tag{3.6}$$

where c is the speed of signal propagation which is close to the speed of light in air.

In the more general case, let $\mathcal{L}$ denote the set of links to a mobile station and $\mathcal{M}$ the set of samples. We can then express the generic sample as:

$$d_{i,m} \qquad i \in \mathcal{L}, m \in \mathcal{M}.$$

Once M samples have been collected for a link $i \in \mathcal{L}$, the distance $\widehat{d_i}$ between the local station and the mobile device is estimated as follows:

$$\widehat{d_i} = f(d_{i,1}, d_{i,2}, \ldots, d_{i,M}), \quad i \in \mathcal{L} \tag{3.7}$$

where f is an estimator of the distance that aims at filtering out the noise of individual measurements. The major contribution of this work is *the development of a robust estimator f that estimates the distance to a target device using WiFi ToF, in the presence of rich multipath as found in common indoor environments, and severe noise as found in commodity WiFi chipsets.*

3.3. Noise Analysis of WiFi ToF-based Ranges

While the firmware-based approach we introduce later in Section 3.5 can allow us to measure the ToF with the best precision using commoditized WiFi radios, the ToF measurements are still affected by large noise coming from severe sources of noise which we discuss in the following in more details. In order to collect samples $\{d_{i,m}\}$ for this analysis, we first determine the reference value $t_{MEAS}(d=0)$ (cf. eq. (3.5) and eq. (3.6)). To this end, we directly connect the AP operating as ToF measuring station to a device to be calibrated and perform reference measurements

[1] A sample is collected per 802.11 DATA/ACK handshake.

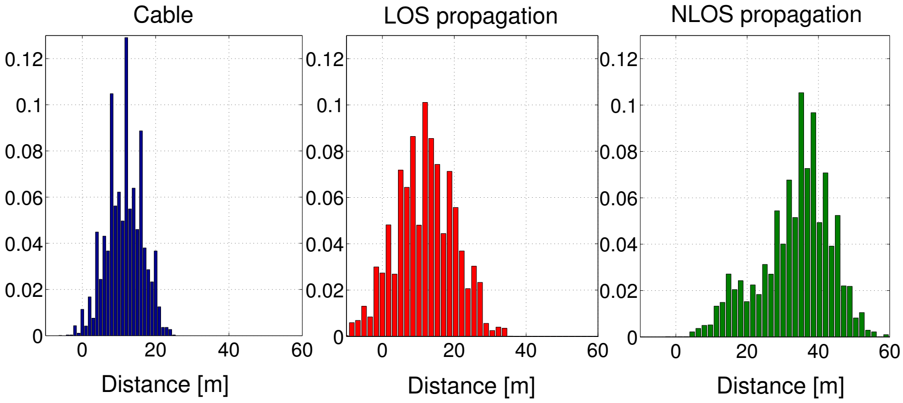


Figure 3.2: Noise introduced by ToF ranging. Tests in a controlled environment (Cable) show that there are sources of large dispersion in the estimation. Tests over the air (LOS and NLOS propagation) show that the distribution greatly depends on the channel conditions. All the tests are in addition subjected to quantization noise and other spurious noise sources.

over three coaxial cables of different lengths. We then infer $t_{MEAS}(d = 0)$ for the distance at 0 m by means of linear regression. This process is required only once per chipset.

3.3.1. Target ACK delay

The 802.11 standard specifies the SIFS time between the reception of a DATA and the transmission of an ACK at the receiver as a fixed interval. In 802.11b, for example, this time is specified as $10\,\mu s$ [4]. However a relatively high tolerance of 1 μs is tolerated which can result in significant noise and distance estimation errors up to 300 meters if the target would fully exploit this specified tolerance level. While most chipsets may not fully exploit this tolerance, the dispersion is still quite significant. To illustrate this, Fig. 3.2 on the left represents the resulting dispersion of a typical Broadcom WiFi chipset. The shown histogram was obtained by collecting sequences of samples $\{d_{i,m}\}$ according to Eq. (3.6) for $10,000$ packets. To avoid any dispersion from environmental effects, the measurements were performed over a coaxial cable of 13.5 meters. We observe that there are sources of noise that can lead to distance estimations that range from 0 to 25 meters, even under ideal signal propagation conditions such as cables.

3.3.2. Multipath reflections

It is well known that signal propagation in complex indoor environments is subject to multipath effects in which multiple copies of the transmitted signal arrive at the receiver over different reflected paths. It is even possible that the direct component is entirely attenuated and the signal is received only over indirect paths. Since signals that travel over indirect paths will take longer time to arrive at the receiver, they introduce a non-Gaussian error in the distance estimation when considering the time-of-flight. This situation is shown in Fig. 3.2 in the middle and on the right where the same experiment as on the left was repeated but for a line-of-sight (LOS) and non-line-of-sight (NLOS) signal propagation link over omnidirectional antennas. In the LOS experiment, there is a visible connection between the measuring station and the target, while in the NLOS experiment, they are obstructed. The dispersion spans a range of 40 and 60 meters for the LOS and NLOS links, respectively, and the large noise in the measurement can also result in negative $d_{i,m}$ samples (see Fig. 3.2 in the middle), especially when the true distance d_i is relatively small. In addition, the NLOS link shows a non-Gaussian distribution. Multipath effects must therefore be taken into consideration in order not to overestimate the distance when dealing with reflected signal propagation paths. Finally, multipath may also happen in LOS links, and thus a method robust to the propagation conditions must be designed.

3.3.3. Quantization and measurement uncertainty

Off-the-shelf WiFi chipsets have not been designed to provide accurate ToF measurements. A main source of noise comes from the coarse clock resolution of the radios. For example, the Broadcom chipset operates with a reference clock of 88 MHz, corresponding to a maximal distance resolution of 1.7 meters. In addition to this quantization noise, off-the-shelf chipsets introduce all sorts of considerable additional noise. As we could see in the histograms of Fig. 3.2, the shape of the distribution is far from being smooth despite using $10,000$ samples to create the histograms, suggesting that the radios must have some bias when measuring the time. Other sources of noise must therefore also be factored in order to estimate the distance. On the other hand, the effects of the clock drift are negligible. See details in [29].

3.3.4. Congestion and interference

Wireless congestion and interference have no impact on the ACK delay (that is, δ_T) since the 802.11 channel is reserved during the SIFS period to the receiving node (i.e. the target station) as dictated by the 802.11 carrier sense multiple access with collision avoidance (CSMA/CA) protocol [4]. However, collisions occur on the wireless medium, causing data retransmissions. Only acknowledged data frames are considered as valid ToF samples, while unacknowledged frames do not generate any ToF measurement. Since the wireless resources are shared, the presence of congestion can increase the time required to obtain a sufficient number of samples for ranging. We study the problem of wireless congestion in an open space room (to reduce as much as possible the multipath, and focus on the impact of interference), with one link of 19 meters in LOS. In this setup, the AP transmits DATA at PHY rate of 1 Mb/s. The AP computes the distance averaging over the collected ToF samples. We then repeat the ranging measurement for the same link, adding 802.11 traffic from another wireless station that saturates the channel with interfering traffic (UDP traffic of 4 Mb/s) sent at PHY rate of 1 Mb/s. The results are presented in Fig. 3.3. The Figure shows the average of the distance estimation, both in absence and presence of interference. While the long-term ranging accuracy is the same in absence of interference and in presence of interference saturating the channel, the latter causes a longer time to converge to the true distance

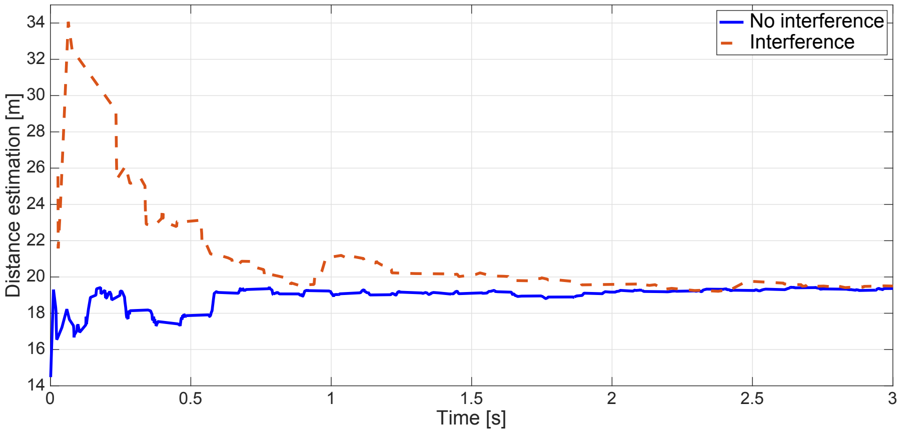


Figure 3.3: ToF distance estimation in a LOS link of 19 meters in absence and presence of interfering traffic. In the experiment, the ToF measuring station sends DATA at PHY rate of 1 Mb/s.

d_i. We can conclude that, in order to efficiently handle interference, it is important to design a system that requires only a few samples M for the ranging estimation.

3.3.5. Device-related Noise Analysis

In this section, we analyze in details the noise originated at the measuring station and at the target station. The goal of this analysis is to quantify the device-related noise and determine whether the non-Gaussian noise we observe in ToF measurements is only due to environmental effects such as multipath or the local and target devices also add non-Gaussian noise to the measured values.

In order to address this question, we use a low-noise oscilloscope to measure the offset distribution of $\{\delta_T\}$. This high-end wideband oscilloscope is an Infiniium 90000A model with a fast sampling rate of 10 GS/s [12]. The internal noise of the oscilloscope can be regarded as negligible compared to the noise introduced by the WiFi radios and therefore provides us a mean to analyze the target offset δ_T in isolation.

On the left of Fig. 3.4, we show the histogram of the target offset δ_T as measured with the oscilloscope using approximately 300 samples, and the resulting Gaussian fitting function. We run the Lilliefors test and find that the hypothesis that the distribution of δ_T is Gaussian *cannot be rejected*, with a high p-value equal to 0.43 while setting the significant level to 0.05. Therefore it is safe to assume that δ_T can be approximated with a Gaussian distribution $\mathcal{N}$ with standard deviation equal to σ_{δ_T}.

We then statistically compare the distribution of $\{\delta_T\}$ measured with the oscilloscope with the distribution $\{t_{MEAS}(d)\}$ measured at the local station with our firmware-based approach presented in Section 3.5. For the comparison, we collect samples of $t_{MEAS}(d)$ in LOS links with limited multipath at $d = \{1, 15, 60\}$ m and draw the quantile-quantile (QQ) plots, a graphical method for comparing two probability distributions. If the distributions are linearly related, the

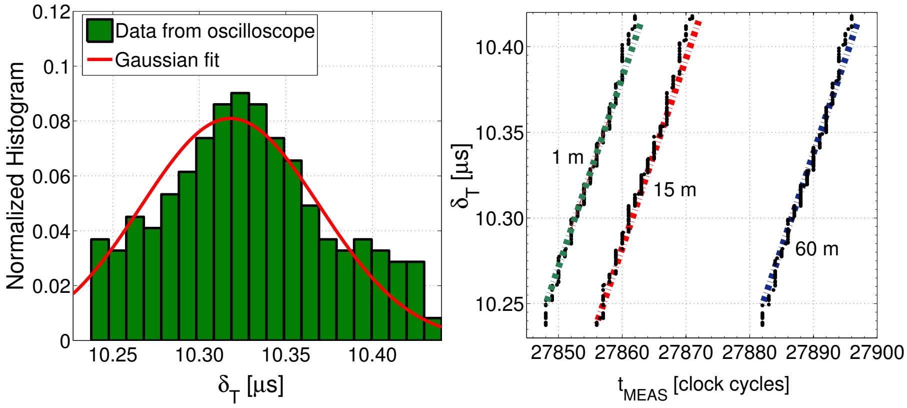


Figure 3.4: On the left: distribution of $\{\delta_T\}$, as measured with our oscilloscope, and its Gaussian fitting function. On the right: QQ plots of δ_T versus $t_{MEAS}(d)$.

points in the QQ plot will approximately follow a line. The resulting QQ plots are shown on the right of Fig. 3.4. We observe a linear pattern, which indicates that $\{\delta_T\}$ *and* $\{t_{MEAS}(d)\}$ *have very similar noise distributions in presence of limited multipath.* We also observe that the dispersion of the two distributions is very similar, with a standard deviation of $4.0 - 4.1$ WiFi clock cycles in both cases. As a result of these investigations, we conclude that:

- The noise of $t_{MEAS}(d)$ in LOS links with limited multipath is largely dominated by the Gaussian noise of the target offset δ_T.

- The dispersion of the local offset δ_L in LOS links has a negligible impact on the distribution of $t_{MEAS}(d)$, and our approach to implement the ToF measurement in the firmware does not add a significant dispersion.

- The non-Gaussian ToF noise that we observe in many indoor links of our testbed is not related to the local or target noise of the WiFi devices but rather to environmental effects such as multipath.

3.4. Dealing with Non-Gaussian Noise

Existing multipath-resistant estimators assume that the non-Gaussian noise from the reflected signal path components are the dominant sources of noise. However, as we have shown in the previous Section, the device-related Gaussian noise is extensive in WiFi radios. In order to deal with these combined sources of noise, the focus of this section is to analyze different robust estimators f to estimate the true distance d for multipath-rich indoor environments.

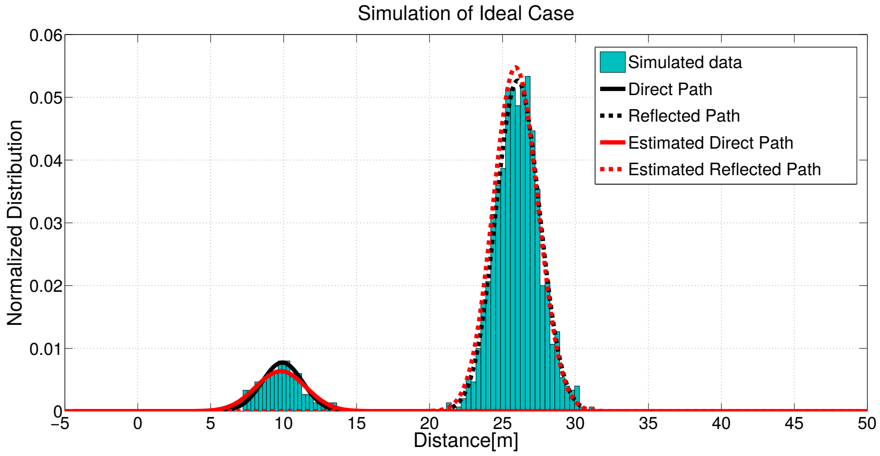


Figure 3.5: Generation of a direct path and a reflected path in simulation using the GMM model, with small Gaussian noise of the target station. As expected, the EM algorithm can reliably estimate the two modes.

3.4.1. Noise Model

While the device-related noise can be approximated as an environmental-independent Gaussian distribution, this is not the case for the noise that arises from the effect of multipath reflections. The amount of multipath is very much dependent on the environment and the exact position of the devices. Under multipath propagation between two WiFi nodes, the distribution of ToF measurements will be non-Gaussian, as a result of those data frames where the 802.11 transceiver synchronizes to a delayed copy of the signal.

Estimating the true distance using this noisy data is not trivial with WiFi radios. In fact, we cannot directly measure the individual multipath components at the signal-level on off-the-shelf WiFi radios since the WiFi chipset only synchronizes to the strongest path, resulting in a positive or null bias $b_{i,m} \geq 0$.

In order to design a reliable estimator f of the distance, we model the ToF noise in presence of multipath rich environment as follows. Let us consider a link $i \in \mathcal{L}$. The true distance d_i of this link is affected by an additive Gaussian noise $\mathcal{N}$, generated by the target station, with the standard deviation equal to σ_{δ_T} (cf. Section 3.3.5). d_i is further affected by a distance bias $b_{i,m} \geq 0$ caused by the absence ($b_{i,m} = 0$) or presence ($b_{i,m} > 0$) of multipath. Therefore, for each sample, we can model the distance $d_{i,m}$ as follows:

$$d_{i,m} = d_i + b_{i,m} + \mathcal{N}(0, \sigma_{\delta_T}), \tag{3.8}$$

where d_i is the true distance.

Grouping together the samples with the same bias after clock quantization (that limits the distance resolution per sample to 1.7 m, cf. Section 3.5), we have a finite Gaussian Mixture Model (GMM) with a small number of modes. One of these modes corresponds to the samples synchronized to the direct path (the samples with $b_{i,m} = 0$), while the others correspond to the samples synchronized to any of the reflected paths. In general, no component is dominant and we can conclude that the Gaussian mixture is non-Gaussian distributed.

3.4.2. Reliability of the Estimation of the Gaussian Components

A fundamental difference with respect to classical GMM applications is that the *multipath noise superimposes to the large noise σ_{δ_T} added by the target device*. The deviation due to σ_{δ_T} can be much stronger than the bias error caused by synchronizing to a delayed copy in a multipath propagation[2]. Classical techniques for learning the modes of the GMM model need instead that the modes are either sufficiently apart or consider that only a small number of samples is received as reflected paths representing a few outliers [35].

In order to verify how well current learning mechanisms for a GMM model could be applied to WiFi ToF, we generate in simulations $1,500$ samples using the GMM model and suppose that there exist only two modes, one mode representing the direct path and one mode representing the reflected path. We then apply the expectation–maximization (EM) algorithm [11] (the details of the algorithm are presented in Section 3.6.3) to cluster the samples. In the scenario, we consider that the direct path has a distance of 10 m and that the reflected path has a traveled distance of 26 m. The mixture weights of the Gaussian components are equal to 0.12 and 0.88, respectively. This indicates that the 802.11 WiFi chipset synchronizes more often to the reflected path, which is in general the stronger signal component. We first make the (ideal) assumption that δ_T is much smaller than in real measurements. We call it ”Simulation of Ideal Case”. In the second scenario, we consider that δ_T is equal to the one empirically measured in Section 3.3.5. We call it ”Simulation of Real Case”.

We plot the results of ”Simulation of Ideal Case” in Fig. 3.5. We observe that the EM algorithm estimates the parameters of the components of the Gaussian mixture almost perfectly: the estimated mean of the direct path's distribution is equal to 9.8 m and the mean of the reflected path's distribution is 25.9 m. It follows that we are able to estimate the distance of the direct path with an error of only 0.2 m. We then plot the results of ”Simulation of Real Case” in Fig. 3.6. Clearly visible from the Figure, we do not have anymore a clear separation of the two Gaussian components. The parameters are here overestimated: the estimated mean of the direct path's distribution is 19.3 m and the mean of the reflected path's distribution is 29.1 m. Concluding, we have a high error in the estimation of the direct (and shortest) path, used for ranging, of 9.3 m.

[2] For instance, in the example in Section 3.3.5, we have a standard deviation of 4 clock cycles of the noise of the target device in LOS conditions. This results in a 99-percentile dispersion of the error of approximately $2.58 * 4(\text{deviation in clocks}) * 2 * 1.7(\text{conversion of clock to ToF distance}) = 35.1$ m, higher than a typical error caused by synchronizing to a delayed copy of the 802.11 signal in a multipath propagation in indoor scenario.

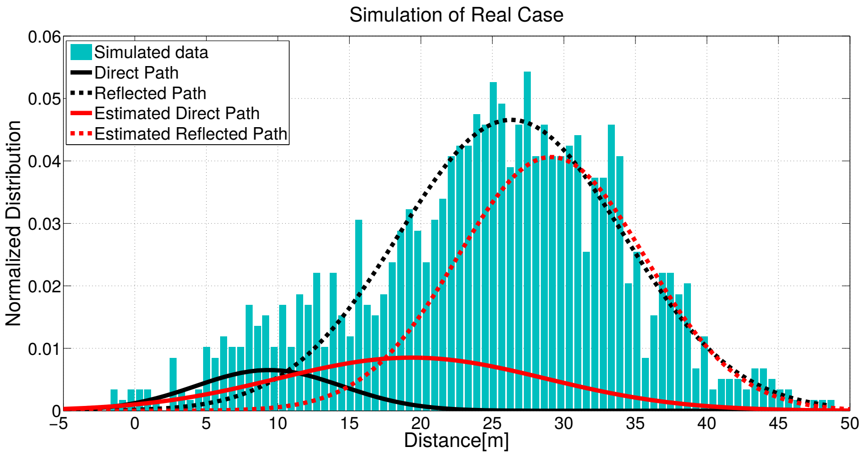


Figure 3.6: Generation of a direct path and a reflected path in simulation using the GMM model, with Gaussian noise of the target station as found in our experiments. The EM algorithm fails to reliably estimate the two modes.

3.5. ToF on COTS WiFi Radios and Testbeds

The design of a robust estimator f requires first to characterize the noise sources of WiFi ToF. For this reason, we implement a ToF ranging system using commodity hardware. To alleviate any unnecessary source of noise or instability from the operating system, ToF measurements have to be performed as close as possible to the radio hardware. Rather than in the driver [29] or upper layers [34], the best method is therefore implementing the code for ToF measurements in the firmware of the WiFi radio chipset. To measure $t_{MEAS}(d)$ in the firmware, we have customized the open-source 802.11 openFWWF firmware[3]. This firmware is written in assembler and runs on off-the-shelf 802.11 Broadcom chipsets, such as the ones widely used in Linksys access points (APs). Our customized firmware reports $t_{MEAS}(d)$ for each successful DATA-ACK frame pair. The timing is regulated by the general purpose timer, running based on the wireless card's internal clock at a rate of 88 MHz. The timer starts to count clock cycles just after the 802.11 processor sets up a register to indicate that a frame has been sent. Once the ACK frame has been received (or the ACK timeout has elapsed), another register gets updated and the timer gets stopped. Every time a measurement is made, the firmware writes $t_{MEAS}(d)$ into a defined address of the shared memory (SHM). The architecture is shown in Fig. 3.7. Since the driver has also access to the shared memory block, it can retrieve the measurement every time an ACK is received[4]. In the driver, we gather additional data about the incoming ACK such as the data rate, the AP MAC address, etc, and store them all in a buffer. Once this buffer is full or a timeout has elapsed, the data is transferred to the user space with the help of UDP sockets. We use UDP sockets since it allows to easily send the data to user-space, and forward it to a central server for further processing

[3] `http://www.ing.unibs.it/openfwwf/`

[4] We operate in promiscuous mode which allows us to know in the driver when an ACK has been received and thus a new data is available in the shared memory block.

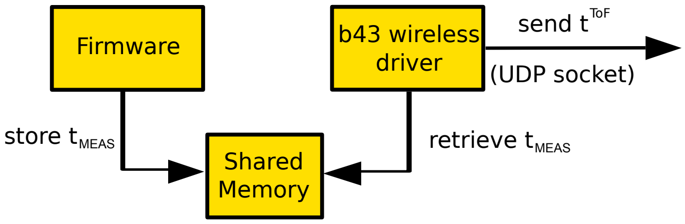


Figure 3.7: TOF measurement architecture. The figure shows the interaction between firmware, driver and user space. Measured ToF data from the firmware is transferred to the driver through shared memory (SHM). Buffered ToF measurements are then sent to user space.

or directly to the central localization unit.

3.5.1. System Setup

We have built a prototype ToF system that consists of COTS APs operating as ToF measuring stations. In our deployments, the AP are static stations. The APs use Soekris net5501 embedded machine with a 500 MHz AMD Geode LX single chip processor. The APs are part of the same 802.11 extended basic service set (ESS) and broadcast the same service set ID (SSID). The APs are equipped with Broadcom AirForce54G 4318 mini PCI type III cards and an omnidirectional antenna. The Broadcom chipset is operated with our customized firmware and the b43 driver presented above. The APs are connected over Ethernet to the location computing unit, that is responsible to process the raw data and compute the position.

As target station we use unmodified Dell Inspiron 5150 laptops equipped with Broadcom AirForce54G 4318 mini PCI type III cards and the integrated antenna of the laptop. In our deployments, the target station may be static or mobile according to the type of experiment (see Section 3.5.2 on deployment scenarios for the details). The target station is associated to the ESS network connecting to one of the APs. At any point in time, the target device is associated to only one AP of the ESS, as in typical 802.11 wireless networks.

In order to perform ranging measurements, the APs use regular DATA frames that are acknowledged by ACK frames from the targets. The ranging measurements are performed in a round-robin fashion among the APs. For every DATA frame, a round trip time $t_{MEAS}(d)$ is measured by the AP from the end of the transmission to the end of the reception of the corresponding ACK (or when a timeout occurs in the case of a loss). Since the target is associated to a single AP, the other APs send their DATA frames with the source MAC address of the AP to which the target is associated. In order to adapt the source MAC address of the APs, we rely raw sockets and the PCAP library[5] which allows us to generate a custom MAC header with any MAC address as the source.

[5] http://www.tcpdump.org/

3.5.2. The WiFi Localization Testbeds

We consider two types of scenarios for the deployment. The first scenario considers controlled experiments, where we avoid any environmental effects by connecting two stations using coaxial cables. The cables are of length $13.5\,\mathrm{m}$ and are based on the standard RG-58. In the second scenario, we perform experiments with transmissions over the air in four indoor testbeds, namely Testbed I, II, III and IV. The maps of the four indoor testbeds are shown in Fig. 3.8. Testbed IV uses the same environment as in Testbed I, but APs are partially placed at different locations.

We deploy 9 APs in Testbed I, 9 APs in Testbed II, 10 APs in Testbed III, and 9 APs in Testbed IV. Testbed I, II and IV are office environments. The environment of Testbed I is shown in Fig. 3.8a. It covers a surface of almost $1000\,\mathrm{m}^2$. We use 25 randomly selected locations (marked as a cross) to test our algorithms. We conducted tests over two different days, with some positions repeated again with different locations of some furniture, and collect a total of 207 wireless links. Testbed II is depicted in Fig. 3.8b. It features 180 links and it covers a smaller space of around $200\,\mathrm{m}^2$. The target station is placed in 20 different positions. Testbed III is shown in Fig. 3.8c. It has been deployed at the facilities of the IEEE/ACM IPSN 2014 - Microsoft Indoor Localization Competition [3]. The testbed has 200 links and it covers $320\,\mathrm{m}^2$. The target device is placed at 20 positions in two rooms and a hallway.

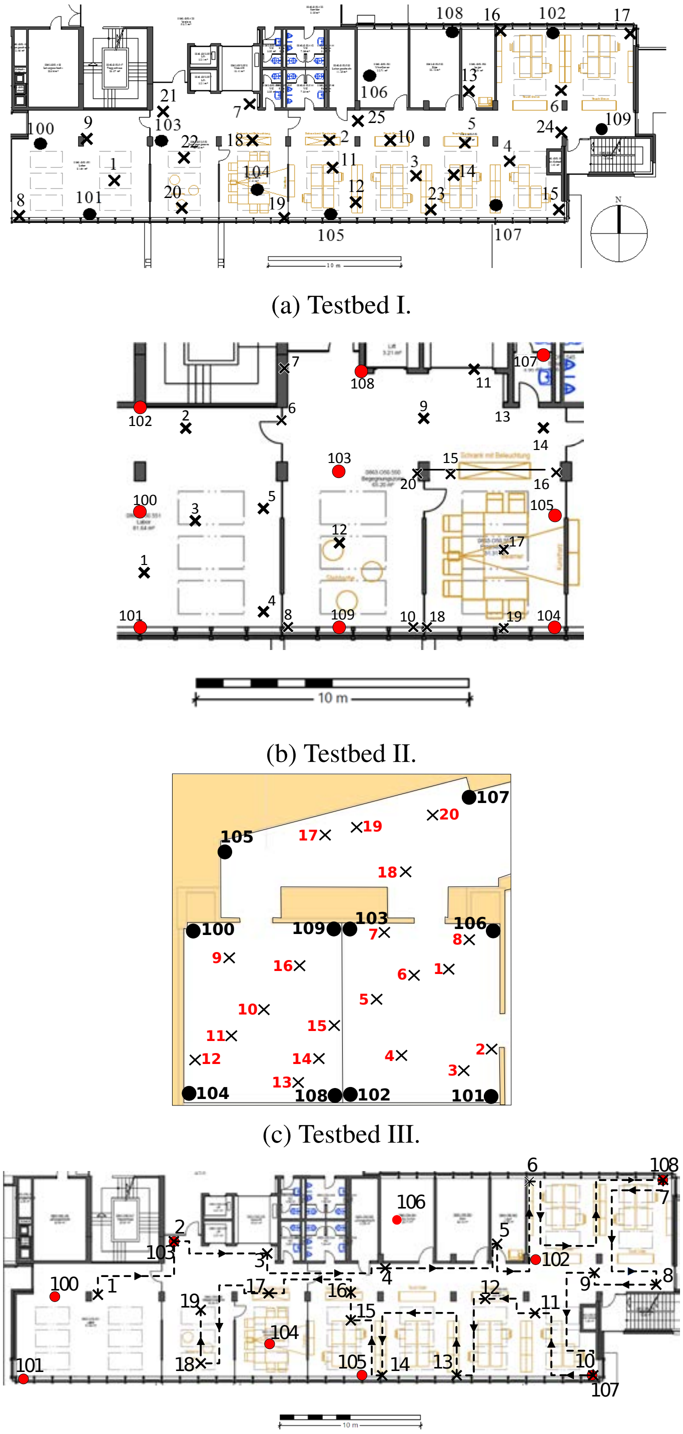
Testbed IV is deployed to study the performance of mobile device tracking. In this testbed, we measure the position when a mobile device is moving from position 1 to 19 as marked in Fig. 3.8d. For these mobility tests, a user moves at a speed of approximately $0.4\,\mathrm{m/s}$. At the time that the user passes at one of the marked positions, we estimate the position with our filter and record the value. We then repeat the tests in the following setups: when the user stops at the marked locations for 2 and 5 seconds, respectively.

In all the testbeds, a mixture of line-of-sight and non-line-of-sight wireless links are present. During all experiments, people are moving within the testbed areas. The testbeds also contain several obstructions, including concrete walls, tables and glasses. All experiments are conducted with other active WiFi networks in the neighborhood. We operate the testbeds on a fixed frequency channel of the 2.4 GHz ISM band. The PHY automatic selection rate is active, such that the measurements include probes sent at different rates.

3.6. ToF shortest Path Estimation: Background

3.6.1. MUSIC Algorithm

First, in order to apply MUSIC to WiFi ToF, we consider that the ToF sequence of samples $\{t_{i,1}^{ToF}, t_{i,2}^{ToF}, \ldots, t_{i,M}^{ToF}\}$ characterizes the wireless link $i \in \mathcal{L}$ between the WiFi AP i and the receiver. We see this sequence as equivalent to the time impulse response of the propagation radio channel. The purpose here is to estimate the P multipath delays; including the first one (the direct path or, in general, the shortest path). So we apply the MUSIC algorithm to



(a) Testbed I.

(b) Testbed II.

(c) Testbed III.

(d) Testbed IV. (same as Testbed I. with a different placement of the APs)

Figure 3.8: Testbeds to assess the ranging and positioning capabilities of our system in static conditions. Circles correspond to AP locations and crosses to target locations.

$\{t_{i,1}^{ToF}, t_{i,2}^{ToF}, \dots, t_{i,M}^{ToF}\}$ in the same way it is applied in [48] to the impulse response. The MUSIC algorithm relies on the eigen-decomposition of the autocorrelation matrix of the frequency domain channel response, which will analogically be in our case the Fast Fourier Transform of $\{t_{i,1}^{ToF}, t_{i,2}^{ToF}, \ldots, t_{i,M}^{ToF}\}$ in N_{FFT} points. In our evaluation we set $N_{FFT} = 100$. The frequency domain signal is then decomposed by projection into two orthogonal subspaces. The signal subspace generated by the eigenvectors corresponding to the P largest eigenvalues of its autocorrelation matrix and the noise subspace generated by the eigenvectors corresponding to the remaining N_{FFT}-P smallest eigenvalues. The problem is reduced finally to finding the P multipath delays that maximize the inverse of the norm of the projection vector into the noise subspace followed by applying a peak detection algorithm to find the positions of the local maxima. The position of the first peak corresponds then to the estimated ToF for the direct path component for distance computation.

3.6.2. CAESAR

Third, CAESAR relies on comparing the standard deviation σ_i of $\{d_{i,1}, d_{i,2}, \ldots, d_{i,M}\}$ to a threshold t_s to decide whether the shortest path component is predominant for the considered link or not. It exploits the fact that the presence of severe multipath results in a higher σ_i with respect to links with only a direct path. So a correction factor γ_s is subtracted to reduce the overestimated distance caused by the multipath. γ_s = 0 if $\sigma_i < t_s$ which means that no correction is needed and $\gamma_s = \sigma_i/2$ if $\sigma_i \geq t_s$ to mitigate the multipath effect on a sequence of samples partially received via a reflected path.

3.6.3. Basic EM Algorithm

Second, the EM algorithm [11] estimates the parameters π_k, μ_k, and σ_k of a Gaussian mixture $g(x) = \sum\limits_{k \in \kappa} \pi_k * g_{\mathcal{N}(\mu_k,\sigma_k)}(x)$, where κ indicates the set of modes. It relies on the iterative maximization of the log-likelihood function weighted by the conditional probability that a considered sample belongs to the mode $k \in \kappa$. In our evaluation we decompose the sequence $\{d_{i,1}, d_{i,2}, \ldots, d_{i,M}\}$ in two modes (with the first one indicating the shortest path)[6] in order to estimate the mean of the first Gaussian distribution, used to compute the distance estimate. The initialization of the parameters π_k, μ_k and σ_k is done as follows: i) $\pi_k = 1/card(\kappa)$, where $card(\kappa)$ refers to the cardinality of the set of modes κ (i.e. we start by assigning equal weights to the Gaussian components of the Gaussian Mixture distribution), ii) for μ_k we randomly select $card(\kappa)$ points from the overall set of data to serve as the initial means, and finally, iii) concerning the initial values of the standard deviations of the Gaussian components, we simply start from the standard deviation of the overall GMM sequence as follows: $\forall k \in \kappa, \sigma_k = \sqrt{var\{d_{i,1}, d_{i,2}, \ldots, d_{i,M}\}}$.

[6] Higher order modes did not show better performance.

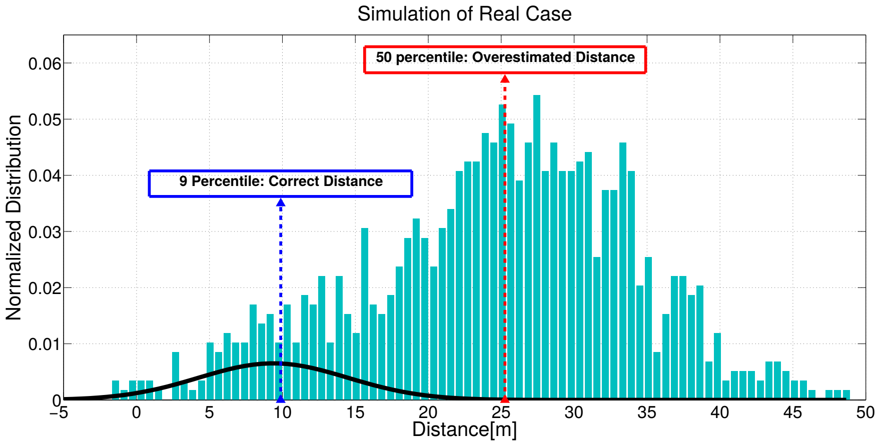


Figure 3.9: Same simulation data set as in Fig. 3.6: We conjecture that, by taking a percentile below 50%, we can counterbalance the biased values $b_{i,m} > 0$ in the estimation process and estimate the mean of the direct path's Gaussian distribution.

3.7. ToF shortest Path Estimation: Adaptive filter design

The results of the previous section have shown that classical algorithms fail to estimate the distance in WiFi echo techniques. In this section, we explore a different methodology for the estimation of the distance. First of all, in WiFi ToF, we are only interested in the direct path's distance or, in general, the shortest path's distance. Indeed, we do not need an algorithm (as the EM algorithm) that estimates the distance of all the paths, including the longer ones. In GMM links with only a direct path ($\forall m \in \mathcal{M}, b_{i,m} = 0$), the median or the mean would be a reliable estimator of the direct path's distance. However, as a result of the GMM model in WiFi ToF, the sequence is in general mixed with samples where the local station synchronizes to a delayed copy of the WiFi signal ($\exists\ m \in \mathcal{M} : b_{i,m} > 0$). In the latter case, using the median of the sequence, e.g. the 50% percentile of the distribution as estimator would result in an over-estimation of the distance. *We then conjecture that, by taking a percentile below* 50%*, we can counterbalance the biased values* $b_{i,m} > 0$ *in the estimation process.* In the example shown in Fig. 3.9, a percentile equal to 9 would be effective to counterbalance the biased value and report the true distance.

Formally, let us consider a link $i \in \mathcal{L}$ and collect a sequence of M measurements $\{d_{i,1}, d_{i,2}, \ldots, d_{i,M}\}$. We define the optimal percentile ${p_i}^{opt}$ as the percentile that provides, for each link i, the minimum absolute distance estimation error with respect to the true distance d_i:

$$ {p_i}^{opt} = \underset{0 \le p \le 0.5}{\text{argmin}}\ |d_i - \widehat{d_i(p)}|, \tag{3.9} $$

where $\widehat{d_i(p)}$ is the estimated distance using the p-percentile of the distribution $\{d_{i,m}\}$. The goal is to design a statistical estimator f that estimates the optimal percentile ${p_i}^{opt}$ based on some observables and gives as output the estimated distance $\widehat{d_i({p_i}^{opt})}$.

Table 3.1: The absolute value of Pearson correlation coefficient ρ between different moments of the RSSI and ToF versus the optimal percentile p_{opt} (0=no correlation, 1=maximal correlation). It is worth noting that the ranging is always performed using ToF measurements. For example, the use of moments of RSSI is limited to infer the optimal percentile, it should not be confused with RSSI-based ranging.

	unfiltered	**pre-filtered**
median of RSSI	0.62	0.63
standard deviation of RSSI	0.04	0.05
skewness of RSSI	0.23	0.24
median of ToF	0.76	0.76
standard deviation of ToF	0.19	0.21
skewness of ToF	0.20	0.51

We train various options for the observables in the estimator f in an extensive evaluation in one of our testbeds (Testbed I, Fig. 3.8a) with all links. As candidate observables, we consider the first three moments (median, standard deviation, and skewness) of the ToF as well as of the received signal strength indicator (RSSI). All three moments could in principle be indicators of multipath. For example, when the median of the ToF is low (or the median of the RSSI is high), the two ranging devices are close to each other and hence likely to have a short line-of-sight connection between each other with little multipath delay. An increased standard deviation of the ToF or the RSSI may indicate that signals are received over several paths with different propagation delays and/or attenuations. The skewness of the ToF and RSSI distributions will also be intuitively larger over links with multiple propagation paths. For each of these six observables, we evaluate two variants, leading to a total of twelve candidate estimators. In the first variant, we determine the moments directly on the raw samples $\{d_{i,1}, d_{i,2}, \ldots, d_{i,M}\}$. In the second variant, we attempt to pre-filter obvious outliers that arise from the device-related measurement noise prior to determining the moments. These outliers are filtered out applying the Thompson Tau technique [75], a statistical method for deciding whether to keep or discard samples based on the expected value and the expected deviation of the sequence of samples.

We evaluate the precision of these estimators by determining their correlation to the a priori known optimal percentile ${p_i}^{opt}$. For this analysis, we compute ${p_i}^{opt}$ for a set of links with known distance, i.e. placing the mobile device at known positions. To quantify the correlation between the different moments and ${p_i}^{opt}$, we rely on the Pearson correlation coefficient [9]. The Pearson correlation coefficient ρ is an indicator of the linear correlation of the variables, where absolute values close to zero indicate a low correlation and absolute values close to one represent a high linear dependence of two variables. A value close to one thus indicates that a moment is a good estimator to predict the percentile that will filter out the multipath noise effectively.

We consider the entire set of links in one of our testbeds (Testbed I). Table 3.1 shows the resulting correlation coefficient ρ for all twelve variants. *The best correlation is provided by the median of the ToF*, followed by the median of the RSSI and the skewness of the ToF. All other

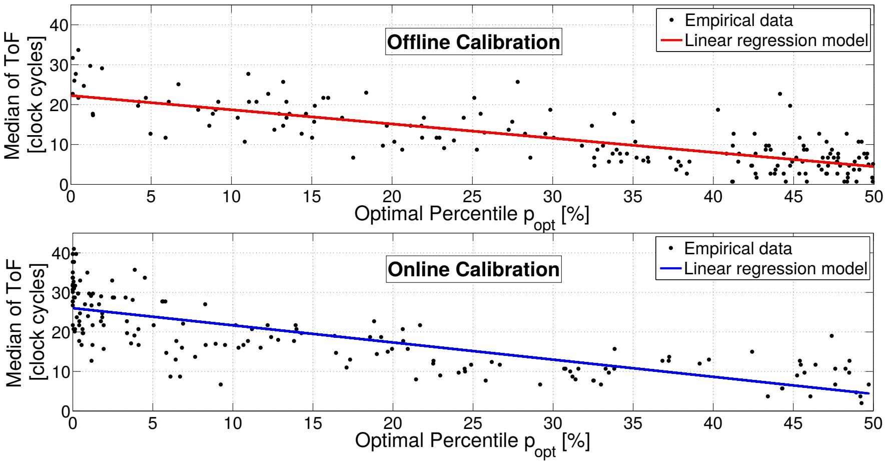


Figure 3.10: Median ToF versus optimal percentile p_{opt} for all links of Testbed I - Linear regression using offline and online calibration

moments have a correlation coefficient below 0.5 which indicates a low correlation. We note that not all moments profit from pre-filtering to remove the outliers. For example, the median of the RSSI and ToF perform worse when outliers are pre-filtered. On the other hand, the skewness of the ToF increases from 0.20 to 0.51 and is therefore considerably better when pre-filtering the outliers.

One may ask why the skewness of the ToF has a worse correlation than the median ToF. Intuitively, the skewness of the distribution should be a good indicator of the multipath, given that links with strong reflected (delayed) components are left-skewed, with p_{opt} smaller than for right-skewed link. Our results suggest that the combined device-related noise of the receiver and the measuring station have a strong negative effect on the correlation on the skewness. This is reflected in the pre-filtered version of the skewness which has a considerably better correlation than the unfiltered version. In contrast, the median of the ToF is much more robust to any device-related noise and therefore outperforms the skewness. The reason for the median ToF working best can be associated to the tendency of having longer reflected paths (and thus longer delays) for links with longer distances and vice versa. In other words, when the ToF is small, the devices are close to each other and the multipath noise will likely not affect much the ToF measurements. On the other hand, when the median ToF is large, the devices are further apart and the links may be affected more severely by the noise from reflected paths.

3.7.1. Proposed Filter Design

Provided the good correlation ρ between the median ToF and the optimal percentile p_i^{opt}, we design a linear model for the estimator f that relies on this correlation to estimate the percentile from ToF measurements. The model is specific to the environment and we therefore train a model for each testbed. The training works as follows. For each environment, we compute the median

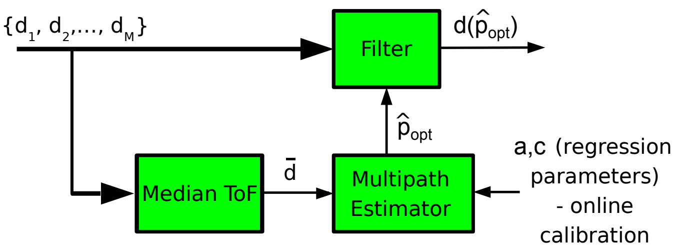


Figure 3.11: Adaptive filter design. A sequence of ToF measurements $\{d_1, d_2, \ldots, d_M\}$ are filtered by selecting a percentile p of the distribution according to the median of the measurements $\bar{d}$ which serves as an estimate of the amount of multipath on the link. The multipath estimator is trained using a linear regression model of the median ToF versus the optimal percentile that minimizes the error from ToF measurements between the APs.

ToF and the corresponding ${p_i}^{opt}$. Note that this requires extensive training at many locations, but we will show how to train a model without manual efforts in the next Section by training using data collected only between the APs. The empirical distribution of the median ToF versus ${p_i}^{opt}$ for all the 207 links of Testbed I is shown in the top of Fig. 3.10.

We note that ${p_i}^{opt}$ is widely distributed between 0 and 50%. Therefore, it does not exist one value of percentile p that it is optimal for all the links, but it rather changes from link to link. Nevertheless, there is a clear linear trend which we will exploit in our estimator f. To get the model, we perform a linear regression on these data points and obtain an environment-dependent linear model, as represented by the continuous line in the figure. Formally, let $\widehat{{p_i}^{opt}}$ be the estimated optimal percentile p_i^{opt} for a link $i \in \mathcal{L}$. $\widehat{{p_i}^{opt}}$ is computed with the following linear regression equation:

$$\bar{d_i} = a \cdot \widehat{{p_i}^{opt}} + c \quad a \leq 0, c \geq 0 \tag{3.10}$$

where $\bar{d_i} = d_i(\widehat{p = 0}.5)$ is the median ToF (estimated distance using the median), and the regression parameters a and c are trained by environment calibration on a given set of known distances, minimizing the sum of squared residuals. Applying classical linear regression theory [19], we can then state that a ρ^2 ratio of the total variation of ${p_i}^{opt}$ can be explained by looking at the median ToF $\bar{d_i}$. Small values of $\widehat{{p_i}^{opt}}$ indicate that stronger multipath is statistically expected using the median ToF for the distance estimation, while values closer to $p = 0.5$ indicate smaller multipath delay.

3.7.2. Automatic Model Calibration

In real deployments, it is desirable to avoid any manual offline calibration to estimate the regression parameters a and c of the linear regression model. We therefore propose to run the calibration methodology based on ToF measurements between pairs of access points (APs) from which we know the exact positions (the assumption of known AP positions generally applies

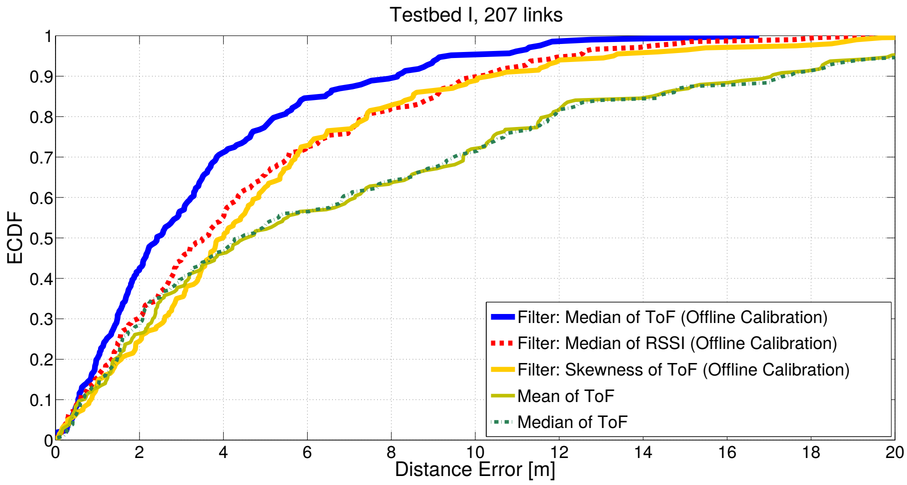


Figure 3.12: ECDF of the distance estimation error for our adaptive filter compared to adaptive filters based on other metrics and classical estimators based on the mean and the median of ToF (Testbed I).

to many localization scenarios such as in airports, malls, museums or companies which have a single wireless network operator.). This can be therefore performed online without human intervention. The results of this online calibration between APs of Testbed I are shown in the bottom of Fig. 3.10. We find that the calibration results in a very similar linear regression. This suggests that *in order to calibrate the model for a particular environment, the ToF measurements between each pair of APs are sufficient*. We illustrate our filter in Fig. 3.11. The individual steps are as follows:

1. We train a linear regression model from online measurements between the APs and determine the parameters a and c of eq. (3.10). The model characterizes the relationship between the median ToF and the optimal percentile ${p_i}^{opt}$ (cf. eq. (3.9)).
2. For each link $i \in \mathcal{L}$, we take a sequence of M measurements $\{d_{i,1}, d_{i,2}, \ldots, d_{i,M}\}$ and determine $\bar{d}_i$ (median ToF).
3. The median ToF $\bar{d}_i$ is used to estimate the multipath delay using the linear regression model from step 1. The output of the estimator is a percentile value $\widehat{{p_i}^{opt}}$.
4. For each link $i \in \mathcal{L}$, we apply a linear interpolation to the sequence of ToF measurements $\{d_{i,1}, d_{i,2}, \ldots, d_{i,M}\}$, select the $\widehat{{p_i}^{opt}}$-percentile of that sequence, and estimate the distance as $d_i(\widehat{{p_i}^{opt}})$.

3.7.3. Evaluation

We analyze the performance of the filter we propose by evaluating the distance estimation error across the aforementioned testbeds.

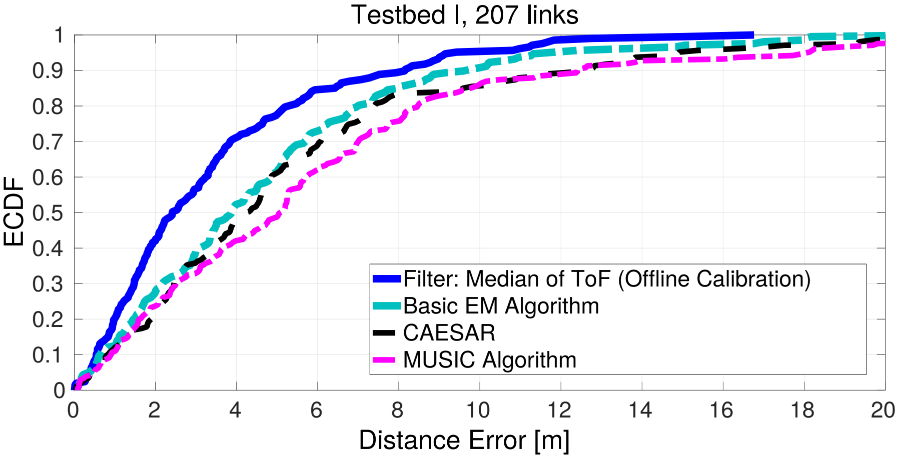


Figure 3.13: ECDF of the distance estimation error for our adaptive filter compared to other algorithms (Testbed I).

3.7.3.1. Distance Ranging Accuracy

We first evaluate the ranging accuracy using all links of Testbed I and offline calibration for our filters (the impact of online calibration is evaluated later). Similar experiments are performed in Testbed II and III. For each link, we compute the distance estimation error with 20 samples, and then calculate the average error using 500 sequences. We consider our filter that uses the median ToF, the median of the RSSI and the skewness of the ToF for estimating the optimal percentile. We also provide the error for CAESAR, the MUSIC algorithm, the GMM model using the EM algorithm, as well as classical estimators such as the mean and median of the ToF.

From Fig. 3.12 we see that our three new estimators outperform the mean and the median metrics by comparing the Empirical Cumulative Distribution Function (ECDF) of the ranging errors. We then compare our best performing filter, based on the median of ToF, against the EM algorithm, CAESAR and the MUSIC algorithm. We show in Figure 3.13 that our filter outperforms the other evaluated approaches. The best performance is achieved by our filter using the median ToF. We obtain a median error of 2.4 m and a 80-percentile error of 5.3 m. The filter that uses the median RSSI slightly outperforms the skewness of the ToF. This is not surprising since Table 3.1 shows higher correlation coefficients of the filter with median RSSI. The mean and median have roughly equal estimation error. Their median error is approximately 4.5 m and the 80-percentile error is approximately 11.5 m. CAESAR gets a median error of 4.1 m and a 80-percentile error of 7.3 m. Our evaluation of CAESAR is also very consistent to the one recently presented in the indoor evaluation of [53]. With regard to the MUSIC algorithm, this approach is ineffective as a result of the large noise introduced by the target station (cf. Section 3.3.5), which is not taken into account in the model used by the MUSIC approach. The 80-percentile of the MUSIC algorithm shows better performance than the mean and the median metrics, but still worse than our new estimators. Finally, the EM algorithm outperforms CAESAR and MUSIC. However, the median error of 3.9 m and an 80-percentile error of 7.1 m shows its inefficiency with

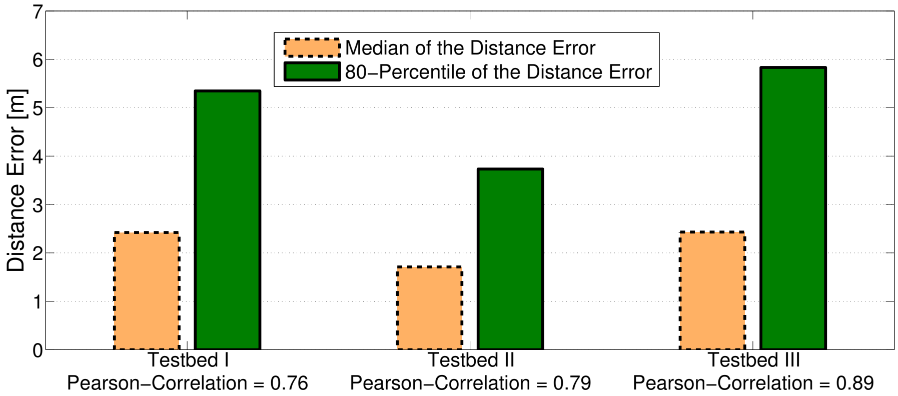


Figure 3.14: Distance accuracy (50- and 80-percentile distance error) in Testbed I, II and III.

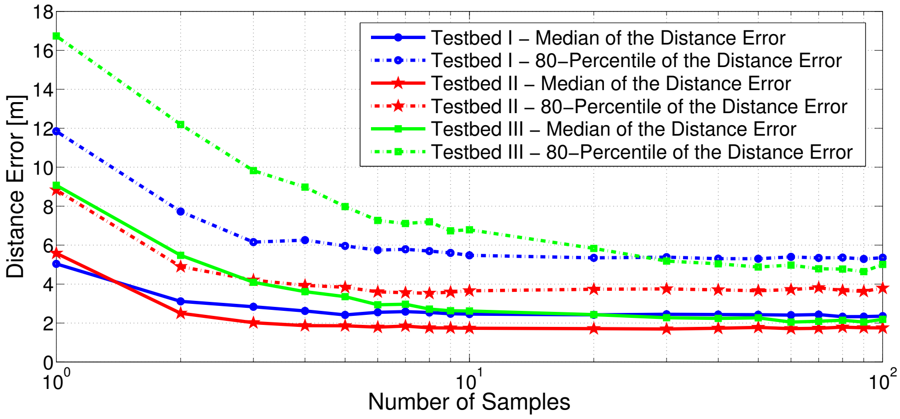


Figure 3.15: Median and 80-percentile error of our estimator that filters the noise based on the median of ToF. Results are provided for different number of samples for Testbed I, II and III.

respect to our filter using the median ToF.

3.7.3.2. Robustness to Different Environments/Testbeds

Fig. 3.14 shows the median and 80-percentile of the distance error for the three different testbeds using sequences of $M = 20$ samples and offline calibration. As shown in the x-label of the figure, we measure a high Pearson correlation coefficient between the median of ToF and the optimal percentile, in the range of $0.76 - 0.89$. The median distance error is in the range $1.7 - 2.4$ m, and the 80-percentile error is in the range $3.7 - 5.8$ m. For comparison, the median (80-percentile) real distances for Testbed I, II and III are equal to 12.3 m (19.2), 8 (11.9) m, 10.2 (15.7) m, respectively. We then examine the relative ranging error. It accounts for the ratio of the ranging error over the real distance separating the AP and the target station. This metric, likewise the previous ones, does not differ significantly along Testbeds I, II and III since the median relative error remains in the range 21.5% - 23.7%.

Further, we show in Fig. 3.16 the distance error versus the true distance in all the evaluated

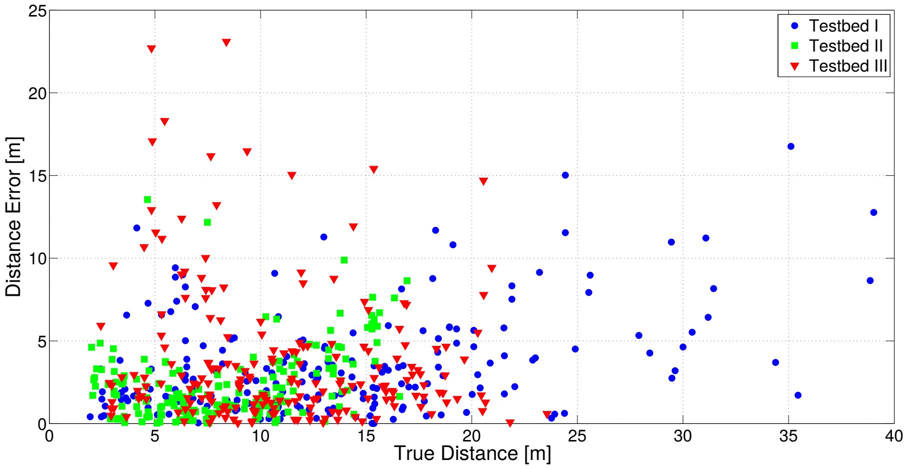


Figure 3.16: Distance error versus the true distance in all the evaluated links in Testbeds I, II and III.

links in Testbeds I, II and III. The correlation between the ranging error and the true distance is 0.20 considering the totality of the links in the 3 testbeds. This shows that the ranging error does not depend on the true distance but it is more related to the existence of a LOS link. There are cases where the true distance is relatively short while the shortest path follows a longer path due to the absence of Line-of-Sight and the multiple reflections that the signal might experience before reaching its destination, and thus a high ranging error. On the other hand, the true distance may be relatively large, but with the existence of a LOS link, the ranging error can be consequently small. Concluding, our filter is robust across different environments.

3.7.3.3. Impact of Number of ToF Samples

We evaluate the number of samples M necessary in our filter. Figure 3.15 shows the error for our filter that relies on the median of the ToF as a function of the number of samples. The error is stable with ten or more samples for the median of the distance error. Only the 80-percentile of Testbed III can benefit from a higher number of samples.

3.7.3.4. Ranging Capacity

We study the capacity of the WiFi ranging technique, defined as the time C required to collect M samples in a WiFi network of N users. In order to find the capacity of the ToF ranging method, we apply Little's formula [71]:

$$C = \frac{M \cdot N}{S/P}, \tag{3.11}$$

where S is the throughput and P indicates the payload bits of the single frame. We conduct the analysis in saturation conditions and we apply the Bianchi's formula to compute S [10]. We also consider that the traffic for ranging does not have data content and it only consists of MAC

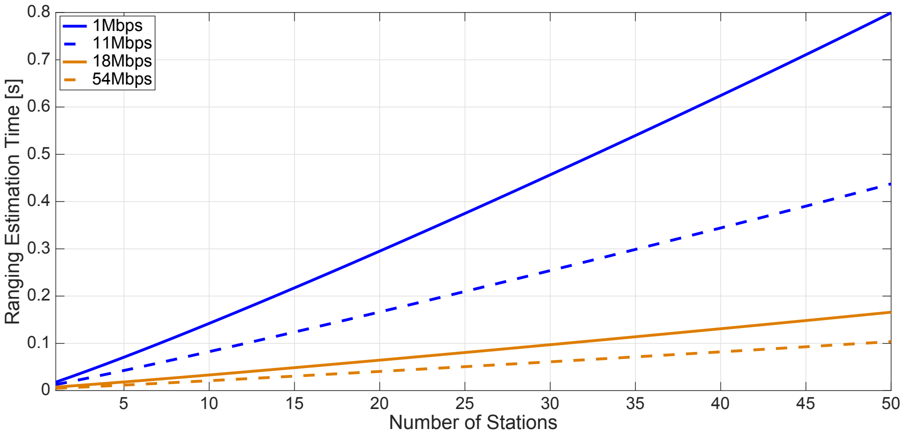


Figure 3.17: Capacity analysis for ranging, considering N target stations and $M = 20$ samples for the ranging estimation.

overhead. Considering $M = 20$ for the ranging estimation, the results versus the number of users in the system are shown in Fig. 3.17. As expected, increasing the PHY rate helps reduce the time C required to compute ranging estimates for N users. For instance, $C = 0.25$ s is needed to compute the ranges to 30 users with PHY rate of 11 Mb/s, while $C = 0.1$ s is needed with PHY rate of 18 Mb/s.

3.7.3.5. Online vs offline calibration

Our proposal for online calibration is to train our model for the adaptive filter using the links between the fixed APs. Since the APs are at fixed known locations, this type of calibration does not require any additional manual effort and can therefore be performed online; indeed, in order to implement a practical tracking system, it is highly beneficial to avoid the need for offline calibration in order to minimize the deployment efforts and costs.

We compare online versus offline model calibration for Testbed I in Fig. 3.18. We have three findings. First, the online calibration achieves similar results with respect to the offline tests, and we can then apply the calibration online without significant performance loss. Second, online calibration outperforms the results we could achieve applying the linear regression achieved in Testbed II and III to Testbed I, which implies that it is better to calibrate online than using the calibration coefficients of other environments. Third, we find that the regression parameters $\{a, c\}$ of Testbed I and Testbed IV are very similar. This is mapped to very similar performance observed in Fig. 3.18 using the offline calibration of Testbed IV. These testbeds use the same environment but different placements of the APs. This result suggests that the calibration is largely AP-position independent, but rather a feature of the environment.

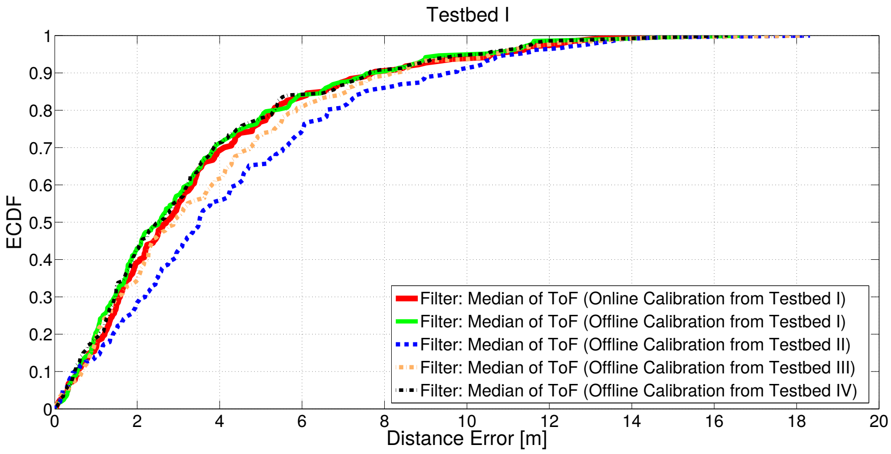


Figure 3.18: Comparison of offline and online calibration for Testbed I, and comparison with different placements of the APs (i.e., Testbed IV).

3.8. ToF shortest Path Estimation: Robust GMM fit

In this section, we propose to investigate a new ranging approach for a shortest path estimator that does not require any offline or online environment calibration, without a substantial loss of accuracy with respect to the adaptive filter presented in Section 3.7. This would greatly benefit pervasive positioning system that suppose no prior environment-related knowledge or to remove this APs-cooperation requirement discussed in the previous section for online environment calibrations needs. Online calibration requires to generate additional wireless traffic, which may be undesired.

In order to address this problem, we consider that, given N measurements, some of them use the Line of Sight (LOS) path (or the shortest Non Line of Sight (NLOS) path in case the LOS path does not exist), and others have one or more NLOS paths. For each single path, we consider a Gaussian distribution for the noise generated by the AP replying with ACKs. Experimental observations of Gaussianity of the single path is demonstrated in Section 3.3.5. We model the sum of all these multipath components as a *Gaussian Mixture Model (GMM)*. A key aspect of the model is to identify the number of dominant paths (clusters) κ, which is up to 5 in indoor environments [44]. We infer the optimal $\kappa \in \{1, \ldots, 5\}$ for the GMM statistical model computing the lowest Akaike Information Criterion (AIC) for the N measurements [42]. We then divide the N measurements in k paths with the GMM components likelihood optimized using the iterative Expectation-Maximization (EM) algorithm initialized by k-means++ [6]. We then compute the means of the k paths.

Our estimator rejects the paths with negative means, as they do not correspond to a physical propagation path, and uses the *path with the least positive mean* as the shortest path to consider for range computation. We use the raw data set collected in Testbed I. and we consider rounds of 20 measurements as we did in Section 3.7 to estimate the distance. We show in Fig. 3.19 how

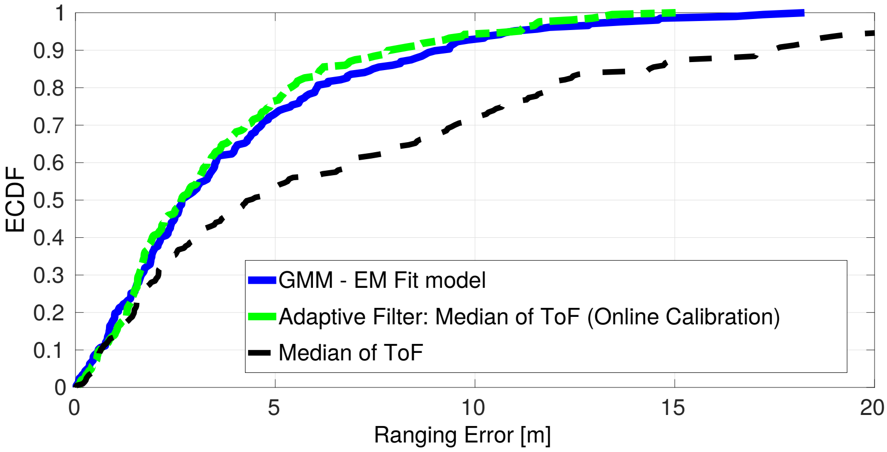


Figure 3.19: Empirical Cumulative Distribution Function of distance estimation using our WiFi ToF GMM model versus the filter described in Section 3.7, 207 links.

close is the ranging error using the GMM fit model to the filter presented in Section 3.7 *without the need of performing any environment calibration*. For this reason, the approach we propose is a valid alternative for systems where it is undesired to generate additional wireless traffic for online calibration. It brings more flexibility with respect to the approach proposed in the previous section while keeping the accuracy at a relatively similar level.

3.9. Wrap-up & Summary

We have demonstrated the feasibility of mobile device tracking using ToF information acquired from COTS WiFi access points and developed a new filter based on statistical learning and robust statistics to improve the ranging accuracy in the presence of noisy ToF measurements. Our estimators do not require additional information form the devices or the user besides the ToF values which makes it applicable to a wide range of applications. We have shown how to apply our estimators for indoor localization of COTS WiFi devices with legacy 802.11 Access Point infrastructures. In indoor deployments with multipath, both our estimators outperform conventional ToF based range estimators by a factor of more than two. We have demonstrated the accuracy of the filter to estimate the position in static and mobile settings. We have shown that the performance of our filter can be achieved with online model calibration, and hence does not require any cumbersome onsite pre-calibration efforts. We further challenged this approach and proposed a shortest path estimator for ToF measurements that does not require any environment calibration and have shown that with respect to the aforementioned adaptive filter, there is no substantial loss in accuracy. This is of a tremendous interest in the following chapter as the autonomy of the system is at the core of the investigation.

Chapter 4

Data Fusion for Hybrid and Autonomous Positioning

4.1. Introduction

The proliferation of handheld devices and the pressing need of location-based services call for precise and accurate ubiquitous geographic mobile positioning that can serve a vast set of devices. Despite the large investments and efforts in academic and industrial communities, a pin-point solution is however still far from reality [52]. Mobile devices mainly rely on the Global Positioning System (GPS) to position themselves, known to perform poorly in dense urban areas and indoor environments where the visibility of GPS satellites is reduced drastically. In order to ensure interoperability between the technologies used indoors and outdoors, a pervasive positioning system should still rely on GPS, yet complemented with technologies that can receive radio signals in indoor scenarios using commercial mobile clients.

Existing mobile devices such as smartphones commonly rely on a multi-RAT (Radio Access Technology). One critical aspect for the localization problem is that the various technologies operate as *monolithic radios* entities. Here the principle is to compute the final positioning selecting the radio technology that is foreseen to provide the highest accuracy [74], with GPS used as a black box with its position reading that could be simply turn on/off [58].

In this chapter, we pose the question of how a client could enrich the set of high-quality ranging measurements for those scenarios with limited number of GPS satellites (*no position can be computed*), or in presence of GPS pseudoranges largely affected by multipath (a rich set of GPS measurements, yet *low accuracy of the position*). We propose to fuse range measurements of diverse radio technologies in order to circumvent the limitations of the individual radio access technologies and improve the overall localization accuracy. The concept of fusion has been a key to achieve reliable position fixes in outdoors with GPS measurements and inertial sensors [5]. In contrast, the literature has only scratched the surface of the problem of how to model and exploit raw ranging measurements from heterogeneous technologies [69].

For practicality of our design, we consider a system that computes the position based on timing information (Time-of-Flight (ToF)) extracted from GPS and WiFi technologies. The latter technology has been selected thanks to its large deployment in indoor areas and widespread usage in mobile devices. Yet, new problems emerge in presence of new and diverse ranges:

Heterogeneous sources of noise. We exploit the fact that accurate ToF is today one of the turnkeys of Global Positioning System (GPS) and we investigate how to bring this concept to a multi-RAT (Radio Access Technology) systems where ranging inputs are made available from diverse technologies for the final positioning calculation. Ranges from diverse technologies are subjected to heterogeneous sources of noise that must be accurately modeled, otherwise making the fusion ineffective.

What ranges measurements to use. A large number of multi-RAT ranges could be available and we question how to select these ranges to achieve both accurate and pervasive position estimates. This diversity can be detrimental in the presence of one or more bad quality measurements which may bias the final position.

WiFi chipset-specific problems. For practicality of the solution, the computation of WiFi ranges must not be hindered by chipset-specific problems, which should be properly removed and solved to achieve an autonomous positioning system.

For this new architecture, our contributions are listed as follows:

- We study EKF-based positioning and how it can be adapted to each technology first by properly setting the right parameters (Section. 4.5).

- We carry out a performance evaluation in terms of positioning accuracy using experimental studies in both outdoor and (multipath-rich) indoor environments to assess the EKF for single-technology multilateration and compare it to the Least Squares approach broadly used in this context.

- We propose the Extended Kalman Filter as a data fuser and we properly model the covariance matrix of the filter. The filter operates in two phases and it can compute at the same time the receiver's position and WiFi chipset bias of new APs in range (Section 4.6). Constrained environments can be successfully addressed with this approach (Sec. 4.8).

- We consider diversity as an asset by detecting and identifying faulty measurements to discard them from the positioning problem, and propose a solution that is more suitable to hybrid localization systems than methods presented in the literature for single technology (Section 4.9).

- We evaluate the system in an indoor testbed using commodity hardware, using a GPS receiver and a WiFi chipset. We collect raw ToF ranging measurements from both technologies. Our system achieves an average accuracy gain equal to 18.75 with respect to pure GPS positioning (when GPS gets a position fix) and 1.38 with respect to pure WiFi positioning (Sec. 4.9.3).

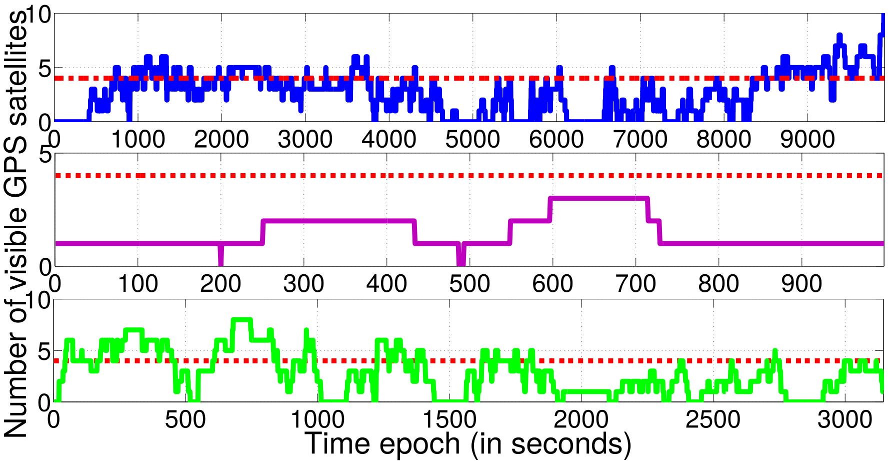


Figure 4.1: GPS availability in (top) Madrid's downtown, (center) PHX international airport and (bottom) the UC3M university campus

4.2. Motivation

We undertake two different types of measurement set to observe the number of visible GPS satellites that can be used for a position fix in key scenarios and how a commercial hybrid solution works in a constrained environment.

4.2.1. GPS availability

We perform a measurement campaign using the U-Center application from U-Blox for Android smartphones. The measurements are stored in a proprietary .ubx format that are further converted to .nmea data format. The raw data collected using a smartphone is not as complete as the one using the U-Blox receiver that we will present in Section 4.3, but it is enough to show the number of GPS satellites in range with the advantage of assessing a real use-case scenario. Fig. 4.1 shows the number of visible GPS satellites (Space Vehicles, SVs) over time observed in three different indoor and urban areas where users may need location-based services: Madrid's downtown, Phoenix (PHX) international airport in the USA and the UC3M (University Carlos III of Madrid) university campus. Figure 4.1 (top) shows that while moving around the Madrid's city center during around 2 hours and 45 minutes, the smartphone's GPS receiver does not receive the minimum SVs measurements required for a position computation all along the 62% of the time. Figure 4.1 (center) reports that while moving inside PHX airport, the GPS receiver does not receive at any time the sufficient number of SVs to compute its position. Figure 4.1 (bottom) demonstrates that the GPS receiver is able to calculate its position only amongst 34% of the test's duration.

These experiments show that pure GPS could fail to provide the user with precise positioning capabilities in indoor and dense urban areas.

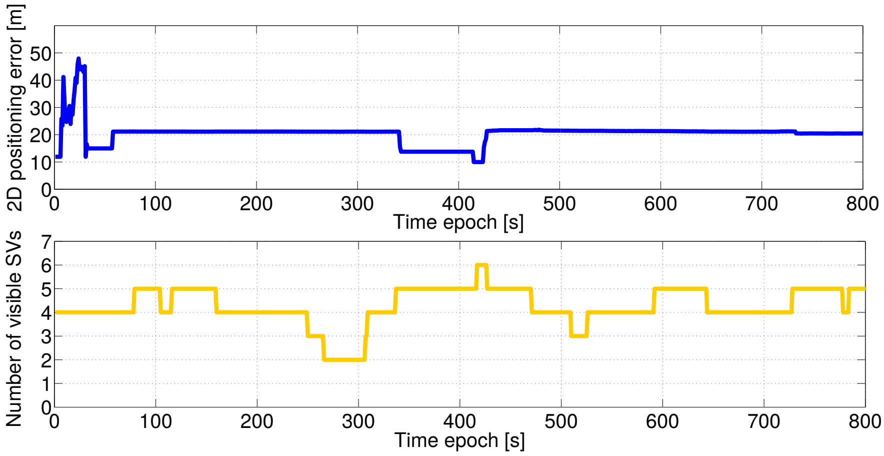


Figure 4.2: The accuracy of the Skyhook Wireless mobile location service.

4.2.2. Hybrid solution in the market

In the same university campus, we test the accuracy of the Skyhook Wireless mobile location service using an Android smartphone. Skyhook is a hybrid positioning system leveraging optimal combinations of WiFi, GPS, and Cellular [1]. The test is performed indoor. While testing it, we deactivate the communication through cellular networks to make sure we use only GPS and WiFi. Figure 4.2 (bottom) shows the number of visible GPS satellites. Being in a university campus, several APs are also in range. We show the accuracy in Fig. 4.2 (top). As GPS requires at least 5 satellites, this positioning system computes the receiver's position in the time intervals when standalone GPS cannot provide it using WiFi APs in range. The best accuracy is achieved in the short time when 6 satellites are available. The root mean square error of Skyhook Wireless computed after convergence of the position estimation in the aforementioned environment is around 20.4 m.

While the position error experienced by commercial solutions suffice to achieve a coarse position estimate, only a seamless position error in the order of meters or even sub-meters can enable novel analytics and new functionalities and services in the network. The main bottleneck in the radio access is that existing mobile devices use the different chipsets as monolithic entities to solve the localization problem. The consequence is that a fine integration of their measurements is not possible. As 5G networks will strongly rely on the concept of multi-RAT (Radio Access Technology), a fine integration in every layer of the stack is needed and in this work we study how to bring this concept to the localization problem, as discussed in the next section and choose the final navigation position from the RAT that is foreseen to provide the highest accuracy in the particular context (see Fig. 4.3 (left)).

[1] `http://www.skyhookwireless.com/products/precision-location`

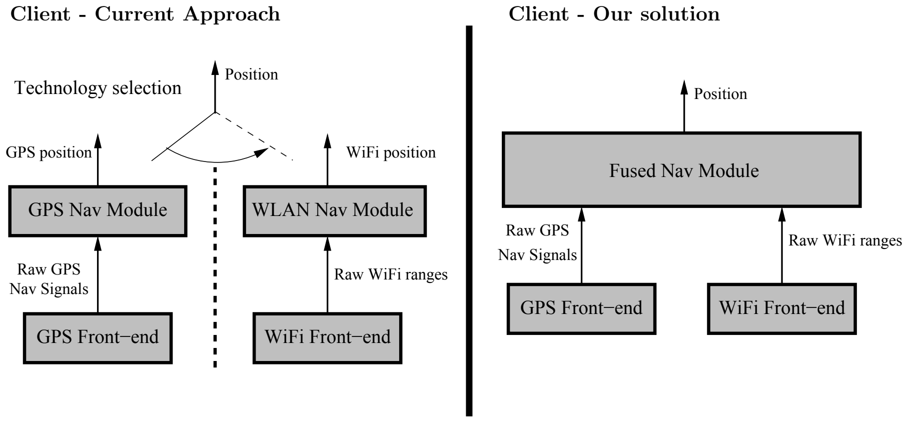


Figure 4.3: High level architecture.

4.3. Evaluation environments

For the evaluation of the proposed system, unless otherwise stated, we place the client in a total of 10 *indoor* positions as shown in Fig. 4.4, where it can receive signals from different SVs (yet at low accuracy) and a total of 5 APs. The environment features a rich-multipath experience with several offices and walls. The measurements were collected at different days and time of the day in order to have distinct GPS satellites constellations and indoor conditions. Ground truth evaluated positions and WiFi AP positions were determined in the LLA (Latitude, Longitude and Altitude) coordinate system using an API to integrate the office's building map to Google Earth and thus determine those positions as precisely as possible. The coordinates are further converted to the ECEF (earth-centered, earth-fixed) coordinate system for homogeneity with the GPS satellites coordinates. Regarding the WiFi hardware, we used exactly the same APs as in Testbed I described more in details in Section 3.5.2 of the previous Chapter. We refer to this main evaluation indoor scenario as *Indoor Scenario #2*. We additionally collected other GPS measurement in another indoor scenario with a reduced number of GPS satellites that we name *Indoor Scenario #1*, and an *Outdoor Scenario* with unobstructed GPS visibility to illustrate better some of the results we present later and prove their robustness across different evaluation scenarios.

GPS measurements. We collect real traces using the Evaluation Kit with Precision Timing manufactured by U-blox and equipped with an active GPS antenna of type u-blox ANN-MS. Data traces contain pseudoranges as well as other parameters such as satellite clock offset, the ephemeris data broadcasted by the satellites that includes the SV clock bias, drift and drift rate, the Keplerian parameters [55], perturbation parameters, etc. More details concerning the Matlab GPS simulator we implemented can be found in the Appendix.

WiFi measurements. For WiFi ranging, we use the ToF two-way ranging based on regular 802.11 DATA frames as explained in details in Chapter 3. The approach that we consider is readily suitable for scenarios where the firmware only provides the number of WiFi clock cycles

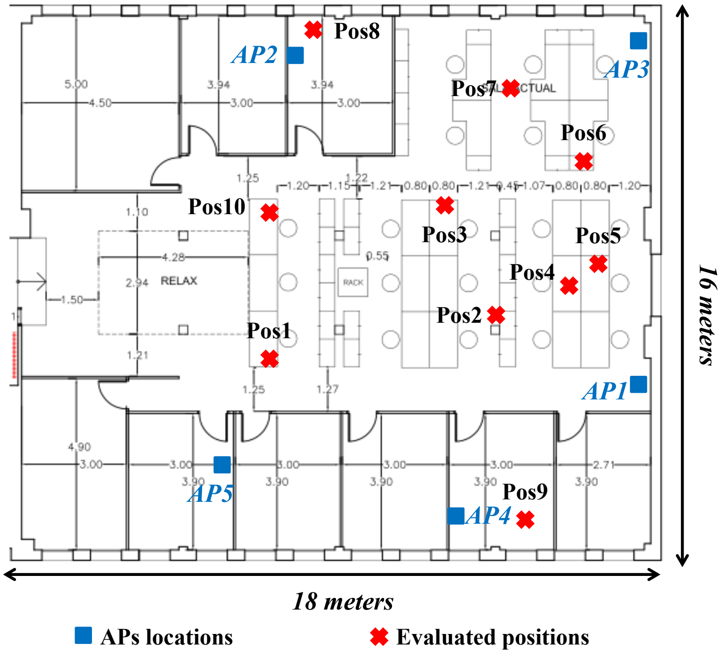


Figure 4.4: Testbed map.

t_{ToF} between the end of the DATA transmission and the end of the ACK reception. We fix σ_{WiFi} equal to 6 clock cycles in our tests given that the clock resolution of the single measurement is equal to the WiFi main clock (88 MHz in this work).

4.4. GPS ToF-based Ranging: Background

WiFi Time-of-Flight ranging theoretical and practical foundations are given in Chapter 3. In this section we recall relevant basics of the GPS Time-of-Flight ranging.

4.4.1. Pseudorange Computation

GPS signals are modulated with a Pseudo-Random Noise (PRN) code unique to each satellite (SV). Each satellite's signal is modulated additionally with a navigation message transmitted in 30 sec. The message consists of 5 frames (of six seconds each) and it includes the ”ephemeris data” and the ”clock corrections and satellite health”, used to calculate the position of each satellite in orbit and compute the distance from each satellite to the client (called pseudorange pr). Each GPS satellite has an on-board atomic clock and includes the timestamp [time sent] of the signal it broadcasts. GPS signals take from 64 to 89 ms to travel from a satellite to the Earth's surface. A GPS receiver also keeps track of time (with its clock) of the received signal [time received].

The receiver first performs a coarse estimation to compute t_{ToF} by comparing [time sent] and [time received] at packet level. The accuracy of this process is in the order of ms. A fine estimation is then obtained computing the sub-millisecond part of t_{ToF} at PRN code level, measuring the time shift t_{ToF} between the local copy and received PRN code. Finally, the pseudorange is estimated as $pr = c \cdot t_{ToF}(d)$.

4.4.2. Correction of GPS Pseudoranges Errors

We give a succinct depiction of the sources of errors in the GPS pseudoranges:

4.4.2.1. Satellite clock offset

The satellite's atomic clock experiences a bias and a drift that must be corrected by the receiver for each satellite it is receiving signal from individually. This correction is approximated by a second order polynomial:

$$C_{SVClock} = a_0 + a_1(t_{tr} - t_{oc}) + a_2(t_{tr} - t_{oc})^2, \tag{4.1}$$

where t_{tr} is the transmission time at the satellite that corresponds to the GPS satellite's time at the receiver t_{GPS} minus the travel time: $t_{tr} = t_{GPS} - pr/c$, and t_{oc} is a GPS reference time available in the ephemeris data. The polynomial coefficients for clock correction a_0, a_1 and a_2 are broadcasted by the satellites in the ephemeris data and denote respectively the clock bias, the clock drift and the clock drift rate.

4.4.2.2. Satellite relativistic effect error

The relativistic effects influence the conversions of clocks proper times at the level of the satellite and on the earth's surface to GPS time and the expression of the positions in a turning geocentric system. The relativistic correction for Earth-Centered Earth-Fixed (ECEF) coordinates and a GPS satellite of eccentricity e and an orbit with semi-major axis A, eccentric anomaly E_k and group delay t_{gd}, is given by:

$$C_{Relativistic} = F \cdot e \cdot \sqrt{A} \cdot sin(E_k) - t_{gd}. \tag{4.2}$$

The inputs parameters are given in the ephemeris data.

4.4.2.3. Ionospheric error

The ionosphere is a dispersive medium ionized by the action of solar radiations, so as GPS signals pass through this atmospheric layer they suffer a delay proportional to the ion density that has a very large spatial and temporal variability, hence the difficulty of modeling the ionospheric time delay. For the implementation, we consider the analytical model developed by

Klobuchar [41] to estimate the correction factor $C_{Ionospheric}$ related to this delay.

4.4.2.4. Tropospheric error

An additional delay is added in the tropospheric layer. We did not correct this error in our implementation because the needed inputs are not available in the raw measurements[2]. Moreover the overall error introduced by this delay in the final distance computation is not more than 0.5 meters [59], which affects very slightly the accuracy of our distance estimation. The tropospheric layer is electrically neutral and non-dispersive for frequencies under 30 GHz. Therefore the effect is identical for both carriers L1 and L2. The tropospheric delay is due to the refraction of GPS signals, the refractive index is function of the temperature, the pressure and the humidity at the receiver's antenna location. This correction is effectuated based on empirical models like Saastamoinen [1973] and Hopfield [1969]. The later is the one widely used according to [reference]. We did not correct this error due to the tropospheric delay in our implementation because the needed inputs are not available in the raw measurements, since they are local metrics they are either gotten from an online server broadcasting this information, which we do not consider in our implementation since we mentioned before that we build a GPS simulator that relies exclusively on data collected by the receiver, or gotten by means of particular sensors that our receiver does not have. Moreover the overall error introduced by this delay in the final distance computation is not more than 0.5 meters, which affects very slightly the accuracy of our distance estimation.

4.4.2.5. Earth-rotation during the travel-time

After using the ephemeris data broadcasted by the satellites to compute their positions at GPS epoch, we have to consider that during the time the GPS signals travel to the receiver at the earth's surface, the earth rotates and consequently the ECEF coordinates system, thus the necessity of transforming the old satellites coordinates. Given a travel time t_{ToF} from a satellite with coordinates X_{SV} to the receiver, the rotated coordinates are:

$$\mathbf{X_{SV}^{Rot}} = \mathbf{M^{Rot}} \cdot \mathbf{X_{SV}}, \tag{4.3}$$

$$\mathbf{M^{Rot}} = \begin{pmatrix} \cos(\Omega_{t_{ToF}}) & \sin(\Omega_{t_{ToF}}) & 0 \\ -\sin(\Omega_{t_{ToF}}) & \cos(\Omega_{t_{ToF}}) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{4.4}$$

where $\Omega_{t_{ToF}} = \omega_{earth} \cdot t_{ToF}$ and $\omega_{earth} = 7.292115 \cdot 10^{-5} rad/s$ is the earth's rotation rate.

[2] They are local metrics they are either gotten from an online server broadcasting this information, or gotten by means of additional sensors.

4.4.2.6. Multipath

Multipath may cause an enlargement of the travel time and thus an overestimation of the range.

Finally the measured pseudoranges from the receiver's antenna to the satellite SV is computed as follows:

$$cr_{SV} = pr_{SV} + c \cdot (C_{SVClock} - C_{Relativistic} - C_{Ionospheric}), \tag{4.5}$$

where cr_{SV} indicates the corrected pseudorange.

4.5. Extended Kalman filter for Single-Technology multilateration

4.5.1. Least Squares-based positioning

Before introducing the Extended Kalman filter positioning, we succinctly present the Least Squares approach widely used for positioning systems as a baseline for evaluation of some concepts among those we study in this chapter.

Given a set of estimated ranges $\{\hat{d}_i\}_{1 \leq i \leq M}$ with M the number of anchors used for positioning, the LS algorithm finds the coordinates $\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ of the mobile client that satisfy the following minimization problem:

$$\hat{\mathbf{p}} = \begin{bmatrix} \hat{x} & \hat{y} & \hat{z} \end{bmatrix}^T = \underset{x,y,z}{\operatorname{argmin}} \sum_{i=1}^{M} (\|\mathbf{x_i} - \mathbf{p}\| - \hat{d}_i)^2, \tag{4.6}$$

where $\mathbf{x_i} = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T$ is the position of the anchor i. We use the Newton-Raphson method for the computation.

For a fair comparison with the EKF presented in the following subsection, we use an Exponentially Weighted Moving Average (EWMA) smoothing of the position fixes estimated by the LS. The reason is that the EKF takes the previous position fix into account to estimate the next one, and then the smoothing property is inherent to this filter. When we refer to LS-positioning in the rest of the chapter, we always mean the LS followed by the EWMA smoothing.

4.5.2. Extended Kalman filter for positioning

We define the state vector to be estimated by the EKF for GPS positioning as $\mathbf{x} = \begin{bmatrix} x & y & z & b_{GPS} \end{bmatrix}^T$ where $\mathbf{p} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ represents the Cartesian coordinates of the receiver, and b_{GPS} the GPS receiver's clock bias in meter unit (multiplying by c) to have an homogeneous state vector. For WiFi positioning, as we suppose the WiFi AP bias known at this level (this assumption is to be eliminated in the next sections), the state vector $\mathbf{x} = \mathbf{p}$. The positioning problem is a discrete-time process with a state model:

$$\mathbf{x_k} = f(\mathbf{x_{k-1}}) + \mathbf{w_{k-1}}, \tag{4.7}$$

and a measurement model:

$$\mathbf{z_k} = g(\mathbf{x_k}) + \mathbf{v_k}, \tag{4.8}$$

f is the process equation that models the kinematic movement equation of the mobile device, while the measurement equations are technology dependent as follows: $g_{GPS}(\mathbf{x}) = \|\mathbf{p} - \mathbf{p_{SV}}\| + ck_B$ for the case of GPS positioning and $g_{WiFi}(\mathbf{x}) = \|\mathbf{p} - \mathbf{p_{AP}}\|$ for WiFi positioning ($\mathbf{p_{SV}}$ and $\mathbf{p_{AP}}$ refer to the coordinates of the position of the anchor considered for ranging measurements, a satellite SV or a WiFi Access Point AP). The usage of the EKF rather than the Kalman Filter (KF) results from the non-linearity of these measurement equations.

$\mathbf{w_k} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q_k})$ and $\mathbf{v_k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R_k})$ represent respectively the process noise and the measurement noise with autocovariance matrices $\mathbf{Q}$ and $\mathbf{R}$. In the next section, we provide an analysis about the measurement autocovariance matrix $\mathbf{R}$ and its impact on the EKF-positioning accuracy.

Adequately characterizing the measurement noise is a decisive step to achieve acceptable performance with the EKF [20]. Given the differences between the two ToF localization technologies we are considering for our analysis, we give hereafter different methods that we use to estimate this measurement noise covariance.

4.5.3. Extended Kalman filter Parameters Tuning

For each technology, GPS and WiFi ToF positioning, we provide in this Section methods and analysis of the characterization of the measurement noise autocovariance matrix $\mathbf{R}$. This step is crucial for a proper tuning of the EKF.

4.5.3.1. Covariance Estimation for GPS-positioning

For GPS, we consider different two ways to estimate $\mathbf{R}_{GPS}$. First, a method that uses a fixed noise covariance and then a method that dynamically estimates this noise covariance. We will provide details regarding both methods before comparing their performances.

Using a fixed value based on HDOP and d_{RMS}

One of the accuracy metrics commonly used to characterize the magnitude of the horizontal error is the Distance Root Mean Square d_{RMS}. It is defined in [40] as: $d_{RMS} = \sigma_{PR} \cdot HDOP$ where HDOP indicates the horizontal dilution of precision (DOP), that relates to the multiplicative effect of the anchor geometry on positional measurement precision error. To capture the localization error intrinsic to the positioning algorithms we are evaluating and avoid including measurement data with high error due to a bad deployment of the AP positions, we calculated the

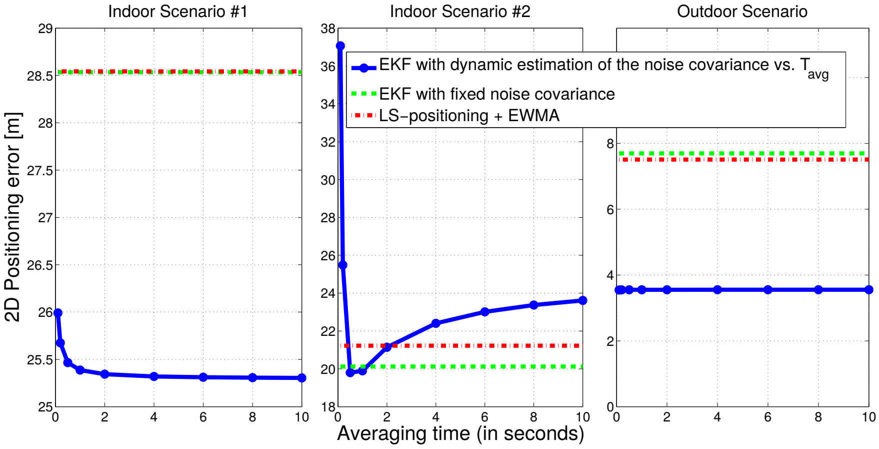


Figure 4.5: The Root Mean Square Error (RMSE) of the positioning versus T_{avg} in different scenarios

horizontal DOP values for all the positions and removed the ones which we considered too high, i.e., we removed the positions with a DOP value higher than five. We did this only for WiFi ToF positioning evaluation since we can have control on the WiFi AP locations but not on the GPS satellites where the DOP is something inherent to the localization process. The probability that the computed receiver's position is not greater than d_{RMS} from the actual position is 0.63. and it ranges between 0.95 and 0.98 within the circle of radius $2 \cdot d_{RMS}$. For our mobile client, we perform several experiments to conclude that the value $d_{RMS} = 8$ m depicts well our receiver since the error falls under 7.81 meters 68% of the time ($7.81 < d_{RMS}$). and under 10.87 meters 95% of the time ($10.87 < 2 \cdot d_{RMS}$) in an outdoor environment. However, when we perform the same analysis in a multipath-rich indoor environment we notice that since the 2D error is higher, the previous value of d_{RMS} does not fulfill the aforementioned accuracy requirements. The d_{RMS} value that fulfills them is $d_{RMS} = 23$ meters instead. This can be explained by the additional bias introduced by the multipath effect. Nevertheless, what is encouraging is that considering the d_{RMS} value got after the tests carried out in the outdoor environment, the overall performance in terms of positioning accuracy is practically the same as with the one proved to achieve the previous accuracy requirements. Therefore, we use the following formula that gives an estimated value of σ_{PR} given the expected accuracy d_{RMS} of the GPS receiver: in a multipath-poor environment, and thus the autocovariance matrix $\mathbf{R}$: Therefore we use the following expression to estimate the standard deviation of the satellites' pseudoranges, and thus the autocovariance matrix $\mathbf{R}$

$$\sigma_{PR} = \frac{d_{RMS}}{HDOP}, \tag{4.9}$$

$$\mathbf{R} = \sigma_{PR}^2 \cdot \mathbf{I_{n_{sv} \times n_{sv}}}, \tag{4.10}$$

n_{sv} is the number of satellites in range, and $\mathbf{I_{n_{sv} \times n_{sv}}} \in \mathcal{M}_{n_{sv},n_{sv}}(\mathbb{R})$ is the identity matrix.

The Early minus Late Delay Lock Loop model

This method takes into account the measurements at every GPS epoch and treats every single satellite separately. The model relies on the structure of the delay lock loop (DLL) used by most GPS receivers, including the one we are using in our experiments, that is based on the correlation of the received GPS signal with slightly early and late versions of that signal. Based on this functioning of the DLL, [55] derived the following formula to compute σ_{PR} of a given satellite $s_i \in \mathcal{SV}, \mathcal{SV} = \{s_{i1}, s_{i2}, \ldots, s_{im}\}$ being the set of visible satellites at the considered GPS epoch:

$$\sigma_{PR}(s_i) = c \cdot T_c \cdot \sqrt{\frac{d}{4 \cdot T_{avg} \cdot \{C/N_0\}(s_i)}}, \tag{4.11}$$

- d: the correlator spacing. It corresponds to the fixed time between the early and late correlator samples. We use $d = 0.1$, as implemented by many receiver manufacturers.

- T_c: the chip time set to $T_c = 1\,\mu$s as in normal GPS receivers.

- T_{avg}: the averaging time. We experimentally derive T_{avg} used in our chipset considering different scenarios (outdoor and indoor) and testing different values of T_{avg}. The results are reported in Fig. 4.5 and show that $T_{avg} = 1$ s provides robust results.

- $\{C/N_0\}(s_i)$ (in dB-Hz): the carrier-to-noise-density ratio from satellite s_i.

The autocovariance matrix can then be written as:

$$\mathbf{R} = \begin{pmatrix} \sigma_{PR}(s_{i1}) & 0 & \ldots & 0 \\ 0 & \sigma_{PR}(s_{i2}) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \sigma_{PR}(s_{im}) \end{pmatrix}, \tag{4.12}$$

4.5.3.2. Covariance Estimation for WiFi-positioning

Also in WiFi, we consider two methods for the covariance estimation. In WiFi positioning, the anchors are in fixed locations which makes the estimation of $\mathbf{R}$ much easier than in the case of GPS where the satellites are permanently moving.

Estimation using a receiver-dependent fixed std-value

The standard deviation of the ToF measurements is lower bounded by the large noise added by δ. We experimentally measure that this noise is in the order of $[4.0, 4.1]$ WiFi clock cycles[3].

[3] Considering that the WiFi clock in our mobile client runs at 88 MHz, the clock cycle is equal to $\frac{10^{6}}{88} \approx 1.13 \cdot 10^{-8}$ sec.

In practice, the multipath adds additional noise in the system. Consequently, we fix $\sigma_{WiFi} = 5.0$ clock cycles and have $\mathbf{R_{WiFi}} = \sigma^2_{WiFi} \cdot \mathbf{I_{n \times n}}$, with n being the number of WiFi APs used for ranging.

The Covariance Matching method for a fixed number of APs

[5] presented an adaptive estimation of $\mathbf{R}$ using the covariance matching principle. It makes the residuals $\hat{\nu}_k = \mathbf{z_k} - g(\mathbf{\hat{x}_k})$ that are the differences between the observation $\mathbf{z_k}$ and its corrected estimated value $g(\mathbf{\hat{x}_k})$ consistent with their theoretical values ($\mathbf{\hat{x}_k}$ is the corrected state vector). $\mathbf{R_k}$ is then computed based on the estimated variance-covariance matrix of the residuals sequences $\mathbf{C_k^{\hat{\nu}}}$ over a moving window of m time instants.

$$\mathbf{C_k^{\hat{\nu}}} = \frac{1}{m} \sum_{i=1}^{m} \hat{\nu}_{k-i} \cdot \hat{\nu}_{k-i}^T \tag{4.13}$$

We only apply this method to WiFi since it requires a stable set of anchors for a sufficient number of measurements, something we do not observe in GPS environments that are affected by the movement of the satellites and consequent variation of the ranging quality.

[5] presented an adaptive estimation of $\mathbf{R}$ using the covariance matching principle. It makes the elements of the innovation $\bar{\nu}_k = \mathbf{z_k} - g(\mathbf{\bar{x}_k})$ or residuals $\hat{\nu}_k = \mathbf{z_k} - g(\mathbf{\hat{x}_k})$ that are respectively the difference between the obeservation $\mathbf{z_k}$ and its predicted value $g(\mathbf{\bar{x}_k})$ or its corrected estimated value $g(\mathbf{\hat{x}_k})$ consitent with their theoretical values. $\mathbf{\bar{x}_k}$ is the predicted state vector and $\mathbf{\hat{x}_k}$ is the corrected one. $\mathbf{R_k}$ is then computed based on the estimated variance-covariance matrix of the innovation sequence $\mathbf{C_k^{\bar{\nu}}}$ over a moving window of m epochs.

4.5.4. EKF vs LS single positioning: Experimental Results

In this section we present the results of the experimental studies we carried out.

Experimental study: GPS ToF localization. The top of Fig. 4.6 shows the 2D-accuracy over time of GPS localization. We observe that the EKF with a dynamic estimation of the measurement noise autocovariance matrix achieves the best performance with a median positioning error of 3.02 m. Using the aforementioned version of the EKF, we gain more than 4 m with respect to the EKF using the same value of the variance of the pseudoranges for all the satellites, that has a median error of 7.39 m. The reason is that a dynamic estimation of the measurement noise is a more accurate model, that considers at the same time the properties of the receiver and the quality of every single satellite where the GPS signals are coming from. The LS-positioning algorithm has a median error of 6.95 m, and it slightly outperforms the EKF-positioning using a fixed value for the noise covariance estimation. Yet the EKF-version with the dynamic estimation of the noise is significantly better. Indeed, when it is well tuned, the EKF performs better than the LS in

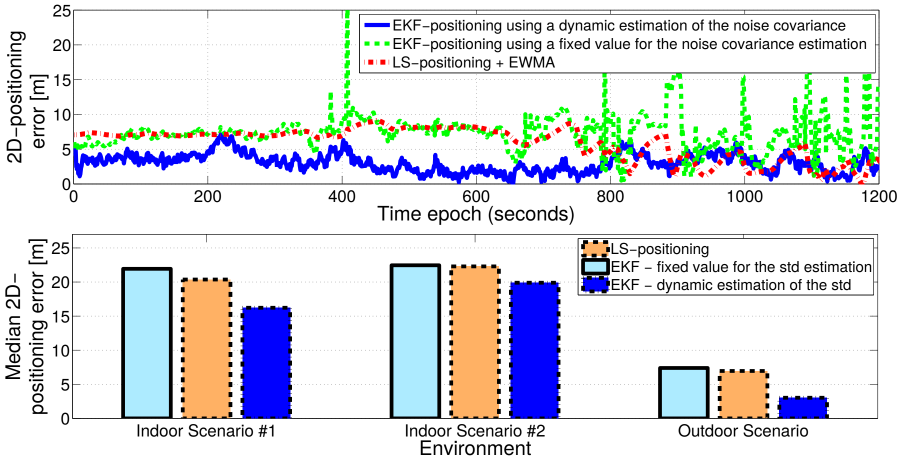


Figure 4.6: On the top of the figure we provide a comparison of the accuracy of different GPS positioning techniques in outdoor scenario, on the bottom we show the median accuracy of GPS positioning techniques in different scenarios.

estimating a discrete-time dynamic process, which is the case here: although we are considering a static GPS receiver, the motion of the satellites makes the whole system dynamic. Fig. 4.5 supports these results while considering different scenarios, for instance an indoor environment with a very reduced SV visibility (Indoor Scenario $\#1$), an indoor environment with acceptable SV visibility (Indoor Scenario $\#2$), and an outdoor environment. Finally, on the bottom of Fig. 4.6, we compare the median accuracy of GPS positioning techniques in different outdoor and indoor scenarios that confirm that we encounter high errors in the indoor scenarios.

Experimental study: WiFi ToF localization. In Fig. 4.7, we compare the accuracies of the EKF and LS based positioning taking as a baseline the positioning approach that takes the nearest AP's position in terms of shortest ToF as the mobile station's position. We evaluate the EKF under different schemes: i) EKF where the inputs are sequences of t_{ToF} obtained after a calibration using the true offset value and ii) under different noise autocovariances matrices, one using a fixed standard deviation (std) value of the noise and the other one using the covariance matching method explained in Section 4.5.3.2, and iii) EKF having as inputs the sequences of t_{MEAS} calibrated using the estimated offset as described in Section 4.7.1 with a fixed autocovariance matrix. For the LS, we only use calibrated measurements using the true offset. Overall, the EKF and the LS have similar performance (median error of 4 meters for the EKF and 4.26 meters for the LS). This is unlike the GPS localization, where the EKF clearly outperforms the LS. The reason for this different result with respect to GPS stands in the ranging computation, where an adaptive filter is needed in WiFi to correct the impact of large indoor multipath. Finally, LS and EKF both perform better than the ”Nearest AP's position” approach with a gain of the median error of more than 2 m.

Using the covariance matching method for the EKF instead of a fixed autocovariance ma-

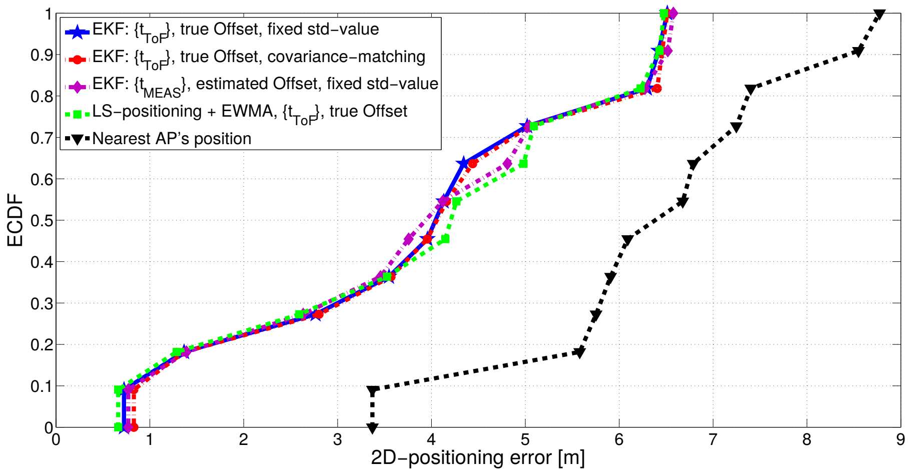


Figure 4.7: Comparison of the accuracy of different WiFi positioning techniques for different positioning algorithms.

trix did not improve the positioning accuracy (both schemes achieve practically the same performance). In fact, we are evaluating a static scenario, wherein the variance of the measurement noise does not considerably change. In this setting, the autocovariance matrix is sufficiently well characterized by the standard deviation value presented in Sec. 4.5.3.2. However, in dynamic scenarios, the number of anchors will change frequently, and we foresee that this will cause problems of convergence of the covariance matching method. Finally, comparing the positioning accuracy between using the true offset versus the estimated one for the chipset calibration, we observe in Fig. 4.7 that using the estimated one we do not have a noticeable loss of performance in terms of positioning accuracy which confirms the robustness of the KF algorithm presented in Section 4.7.1.

4.6. EKF as Data Fuser

In this section, we assume that the client opportunistically exploits any available anchors (GPS satellites and WiFi APs in the study) and infers the ranges to these anchors to compute its position. A high-level representation of the system with ranges from GPS and WiFi technologies is shown in Fig. 4.8.

We compute ToF ranges using WiFi and GPS technologies. A linear relation holds between the ToF (propagation time) t_{ToF} of radio signals and the distance d, $d = c \cdot t_{\mathrm{ToF}}(d)$, where c indicates the speed of light. Ranges (regardless of the technology) are *computed by the device* that aims to pervasively position itself. ToF ranging requires one way ranging for GPS. We use two-way ranging for WiFi, which has the advantage of being 802.11 standard-compliant [28].

Our client operates as follows:

(a) It receives the GPS signals from satellites (SVs) and sends DATA frames to WiFi APs in communication range.

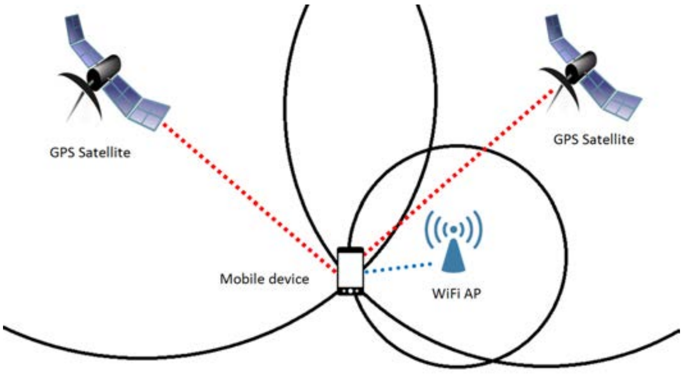


Figure 4.8: Positioning with ranges from different technologies.

(b) It estimates the ToF from the SVs and APs.

(c) It computes its position with a multilateration multi-technology algorithm. Computation of the position could be also partially offloaded to the cloud.

4.6.1. Data fuser for positioning

When the total number of ranging measurements of the different technologies is higher than the number of variables to estimate, the location problem can be posed as an optimization problem. We opt for the Extended Kalman filter (EKF) as a data fuser for the raw ranges as it gives us the possibility to treat any measurement differently, taking into account factors related to the technology used for the specific measurement. Using the EKF fuser, we express the positioning problem as a discrete-time process with state model $\mathbf{x_k} = f(\mathbf{x_{k-1}}) + \mathbf{w_{k-1}}$, and measurement model $\mathbf{z_k} = g(\mathbf{x_k}) + \mathbf{v_k}$ where $\mathbf{x_k}$ is the state vector, $\mathbf{z_k}$ the measurement vector, and $\mathbf{w_k} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q_k})$ and $\mathbf{v_k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R_k})$ represent, respectively, the process noise and the measurement noise with autocovariance matrices $\mathbf{Q}$ and $\mathbf{R}$.

4.6.2. Measurement model

Both GPS and WiFi are affected by a measurement bias:

- GPS bias: it is caused by the clock of the receiver, and it is then client dependent (b_{GPS}). The bias is expressed in meter unit (to have a homogeneous state vector) multiplying the time measurement by the speed of light c. The bias is also subject to the drift d_{GPS} in meter/sec, and thus it must be *estimated continuously* [55].

- WiFi bias: as mentioned in Section 3.3.5, the WiFi bias is caused by the delay of the receiver to schedule the ACK. Therefore, we have a vector $\mathbf{b}_{\text{WiFi}}$ with size equal to the number of APs in range. This bias is corrupted by large noise [28], but it is not affected by drift. As such, *estimating the bias once is sufficient*.

Using one-way ranging in GPS and two-way ranging in WiFi, we express the measurement function $g(\mathbf{x_k})$ as follows:

$$g(\mathbf{x_k}) = \begin{bmatrix} \{\|\mathbf{p} - \mathbf{p}_{\text{anchor},i}\| + b_{\text{GPS}}\}_{i\in I} \\ \{2\cdot\|\mathbf{p} - \mathbf{p}_{\text{anchor},j}\| + b_{\text{WiFi},j}\}_{j\in J} \end{bmatrix}, \tag{4.14}$$

where $\mathbf{p}_{\text{anchor}}$ refers to the coordinates of the position of the anchor considered for ranging measurements, I indicates the set of satellites and J the set of APs. We assume that the APs position can be fetched from an online database (similarly to Skyhook, Google, and other localization systems). The multiplication factor '2' in WiFi in Eq. (4.14) accounts for the use of two-way ranging.

Model of the autocovariance noise of the measurement. The terms of the measurement autocovariance matrix $\mathbf{R_k}$ corresponding to GPS can be computed considering typical GPS receivers that use the early-minus-late discriminators method [55] for tracking the code delay of the incoming GPS signal. Here, we use the model of the standard deviation for the early-minus-late discriminators described in [55], with raw data collected with our GPS receiver (cf. Sec. 4.3).

For the terms of the measurement autocovariance matrix $\mathbf{R_k}$ corresponding to WiFi, the noise of the measurement is mainly due to the uncertainty added by the AP replying with ACKs, which can be of several clock cycles [28], and by the presence of multipath [53]. Multipath is handled by our ranging filter proposed in Section 3.7. We assume that the noise in the measurement consists only on the noise *on the shortest path* added by the AP replying with ACKs. Taking into account that the range is estimated using N consecutive ToF values, the components of the measurement covariance of WiFi are equal to $\frac{\sigma^2_{WiFi}}{N} \quad \forall j \in J$.

4.7. WiFi chipset bias computation: from fingerprinting to GPS-aided estimation

The standard approach to perform ToF in WiFi is to consider that b_{WiFi} for each AP is calibrated offline using controlled measurements, for instance using cables connected to the WiFi transceiver [29]. However, b_{WiFi} depends on the chipset's manufacturer and the transmission rate. This presents the drawback of calibrating each WiFi chipset. This is unfeasible considering the application scenario of a mobile that aims to measure the ToF to any AP in range to position itself. We first introduce a methodology to estimate b_{WiFi} in an iterative way using fingerprinting relying on the data previously collected for the environment training for pure WiFi positioning (i.e. no additional manual work) and further in technology fusion scenario, we propose to opportunistically take benefit of available GPS ranges to estimate this fixed bias/offset b_{WiFi} while still being able to compute the position fix.

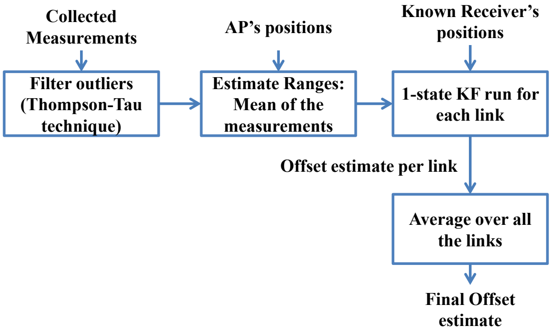

Collected Measurements
AP's positions
Known Receiver's positions
Filter outliers (Thompson-Tau technique)
Estimate Ranges: Mean of the measurements
1-state KF run for each link
Offset estimate per link
Average over all the links
Final Offset estimate

Figure 4.9: Block scheme for the offset estimation

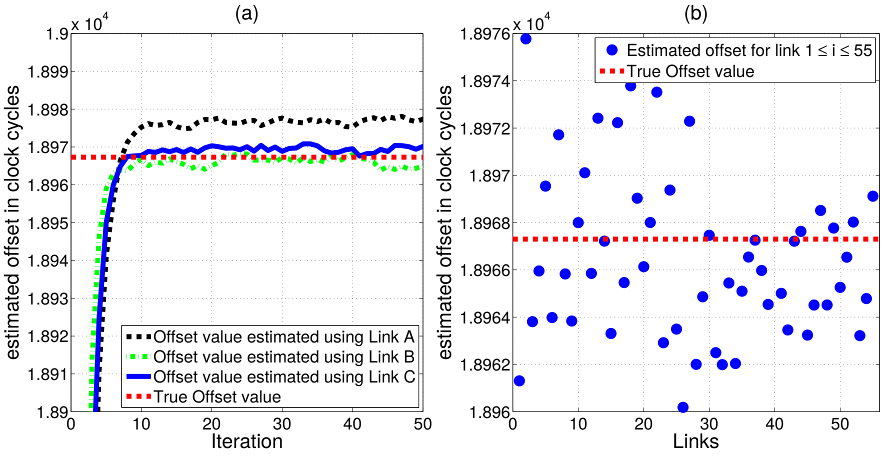


Figure 4.10: Offset estimation for various links. The estimator converges quickly, after approximately 10 iterations (802.11 frames).

4.7.1. WiFi ToF bias estimation: Fingerprinting

Using the Kalman Filter (KF)[4], and following the measurement model of GPS that we presented in Sec. 4.5.2, we observe that there is an analogy between the WiFi ToF measurement offset b_{WiFi} and the clock bias at the GPS receiver side b_{GPS}, since both of them add an offset to the ToF. From eq. (3.5), we can then express the measurement model as:

$$g(\mathbf{x}) = 2 \cdot \|\mathbf{p} - \mathbf{p_{AP}}\| + b_{\mathrm{WiFi}}, \tag{4.15}$$

where b_{WiFi} is equivalent to the b_{GPS} in GPS estimated iteratively in the state vector. The block scheme for the offset estimation is depicted in Fig. 4.9.

For the estimation, we collect $\{t_{MEAS}(d_n)\}_{1 \leq n \leq N}$ sequences for a given set of positions of the mobile device in the indoor testbed (known beforehand), and for a fixed position of the AP in range. This procedure is performed once, as it is also needed for the environmental training of

[4] The eq. (4.15) is linear because the positions of the mobile client are known. Therefore, we use the KF instead of the EKF.

WiFi ranging presented in the previous chapter (cf. Section 3.7).

Since the coordinates (x, y, z) of the mobile client are known, the only remaining unknown is the offset b_{WiFi}. We use the mean of the sequences $\{t_{MEAS}(d_m)\}_{1 \leq m \leq M}$ after that the outliers are filtered using the Thompson-Tau technique to estimate the WiFi ranges that will be the inputs of the KF later on to estimate b_{WiFi}. For the implementation, we use $M = 30$ samples to reliably perform this outliers filtering. As mentioned before, we use a 1-state KF to estimate the b_{WiFi}, which means that only one equation is needed. We then take the average value of all these values for the set of links with the same WiFi chipset in the AP[5] as a final b_{WiFi} estimate. In Fig. 4.10(a) we illustrate the offset estimation for 3 different links to show that the convergence is very fast, considering as initial state the nominal value of b_{WiFi}, given by $\delta = 10\,\mu s$ and t_{ACK} derived based on the transmission rate [70]. The final estimated offset value for a given link is the average over all the iterations after convergence. Additionally, Fig. 4.10(b) shows how the estimated offsets per links are spread around the true offset value, thus the use of the averaging to find the final offset estimate.

Robustness of the offset estimation for different transmission rates. The previous analysis was carried out using the ToF measurements with ACK transmitted at 11 Mb/s transmission rate. We perform the same algorithm using the collected samples with ACK transmitted at 1 Mb/s transmission rate to show that this method gives accurate estimation of b_{WiFi} for different transmission rates. Table 4.1 summarizes the results. We observe that the estimated offset is remarkably close to the true offset, measured in a controlled test using coaxial cables between the WiFi chipsets.

Impact on the environmental training model. We use the environmental training presented in Section 3.7 to achieve a fine estimation of the distance. In this section, we study how robust is the environmental training using an offset b_{WiFi} that is estimated as in Fig. 4.9.

[5] In this work, the APs have the same WiFi chipset. The algorithm works as well for APs with different WiFi chipsets.

Table 4.1: True and estimated offset for the Broadcom AirForce54G 4318 wireless card in the APs.

Rate [Mb/s] (802.11b)	1	11
True Offset [clock cycles]	27854.0	18967.3
Estimated Offset [clock cycles]	27855.6	18966.6

Table 4.2: Coefficients of the linear regression models used for the environment calibration for estimated and true offset values

regression coefficients using true offset	-0.17	11.29
regression coefficients using estimated offset	-0.18	11.68

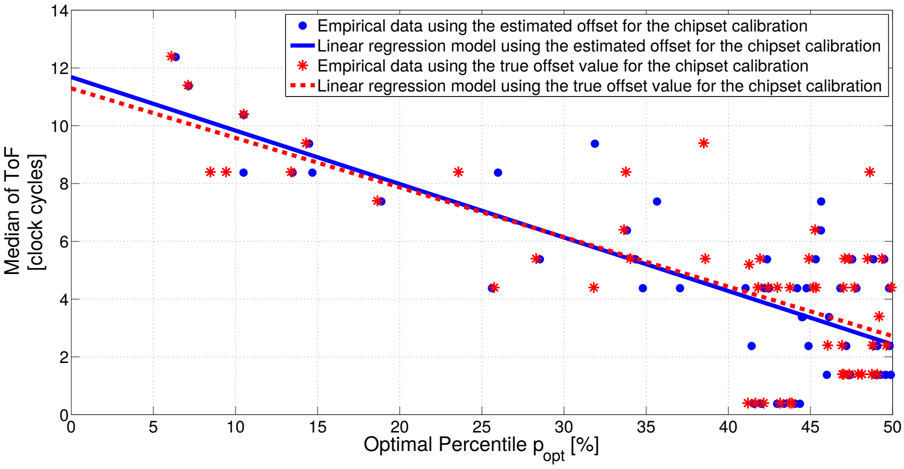


Figure 4.11: Environment training: the linear regression model using the estimated offset does not differ considerably from the one using the true offset.

Since the environment training is defining a linear regression model linking the optimal percentile and the median of sequences of t_{ToF}, this environment training depends in fact on the chipset calibration because it is what allows us to obtain t_{ToF} starting from t_{MEAS} (see Eq. (3.5)). Fig. 4.11 shows that the new environment training model we get using the estimated offset value is close to the one we get using the true offset, which shows the robustness of our approach to calibrate the chipset offset. In Table 4.2 we give the numerical values of these coefficients.

4.7.2. GPS-aided estimation - Two-phases state model

We propose a two-phases model of the state that takes advantage of the fact that the bias of GPS must be estimated continuously while the one of WiFi should be estimated only once. The client first connects to an online database to verify if the AP bias is available. If not, it goes to phase I, where $\mathbf{b}_{\text{WiFi}}$ is unknown and part of the state vector. Here, the client estimates itself the set of WiFi bias $\mathbf{b}_{\text{WiFi}}$. As $\mathbf{b}_{\text{WiFi}}$ depends on the chipset of the AP [28], calibration of each WiFi AP chipset is needed. In fact, every AP may add a bias different from other APs. Once the bias per AP has been estimated, the value is added to the online database. On the other hand, the GPS bias b_{GPS} is added at the level of the GPS receiver as GPS is a one-way Time-of-Flight ranging technology. It is estimated at every time epoch iteratively as a component of the EKF state vector.

Despite there is the very new 802.11mc standard providing the methods to support timing measurements, it is not realistic to suppose that it will be implemented in every AP worldwide in a reasonable time frame. Our method to solve this problem is to focus on scenarios with *only* one AP (AP_j) and enough GPS SVs. For instance, this occurs when the mobile is outdoor with good satellite visibility and AP_j (indoor or outdoor) is in communication range. Under this hypothesis, the state vector is composed by 5 unknowns (x, y, z, b_{GPS} and $b_{\text{WiFi},j}$, where j indicates the specific AP that we have to model) and the EKF filter can be solved using at least 5 ranges (one AP and at least 4 SVs). Once $b_{\text{WiFi},j}$ is estimated (together with the other unknowns), the AP can

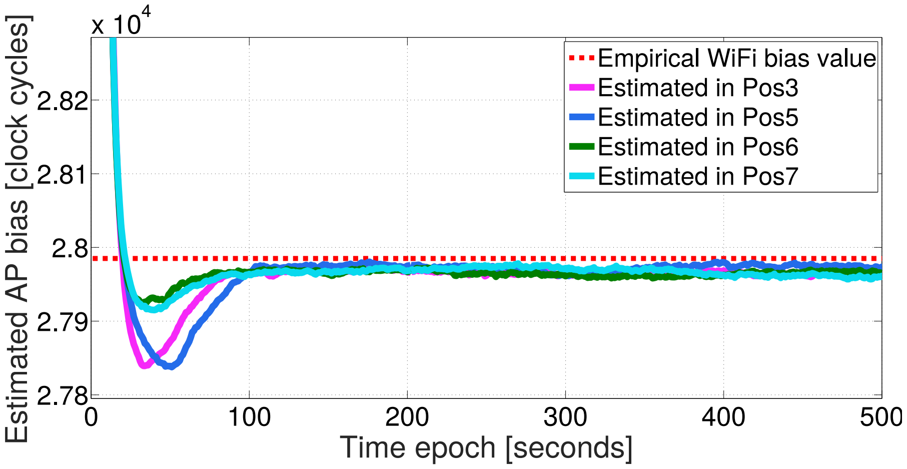


Figure 4.12: Accuracy of the WiFi bias chipset estimation with state model in phase I.

be used for fused positioning (phase II). In phase II, $\mathbf{b}_{\text{WiFi}}$ is removed from the state vector and a state vector with 4 unknowns (x, y, z, b_{GPS}) is used, as in standalone GPS.

Model of the autocovariance noise of the process. For the autocovariance matrix of the process noise $\mathbf{Q_k}$, we use the kinematic equation of movement for the terms of the state vector related to position and velocity, and similarly for the bias and drift. For the WiFi bias in the process noise $\mathbf{Q_k}$, we use the same fixed standard deviation value σ^2_{WiFi} for each AP.

WiFi bias online estimation. We study the accuracy of phase I of the EKF model, considering all available GPS SVs (without removing any range) for position. We have two observations. First, from Fig. 4.12, we notice the robustness of the estimator that shows similar performance regardless of the selected position. Second, the value is close (approximately 2 cycles smaller) to the empirical value measured experimentally in controlled experiments.

4.8. Hybrid Positioning in Constrained Environment

The fusion of ranges as performed by the proposed data fuser allows to locate the client in constrained environments where there are only a few SVs and APs for positioning. For the evaluation, we consider a scenario with only 3 SVs and 2 APs available for positioning. We use only a subset of 2 WiFi APs from the 5 APs deployed in the indoor testbed described in Sec. 4.3 to understand well how the GPS and WiFi range fusion performs in this specific context. As standalone GPS requires at least four SVs for 3D positioning to find the four unknown variables (x, y, z, b_{GPS}) and standalone WiFi requires at least 3 APs for 2D positioning, none of them is capable to locate the device using a traditional multilateration algorithm. The best that can be done with this data set alone is to use the two APs and simply consider the position of the WiFi AP with strongest signal strength as position fix. We show the results in Fig. 4.13 as a function of time. In the same figure, we also show the accuracy achieved fusing ranges of GPS and WiFi with the EKF introduced in Sec. 4.6. We observe that our approach gives better accuracy than a

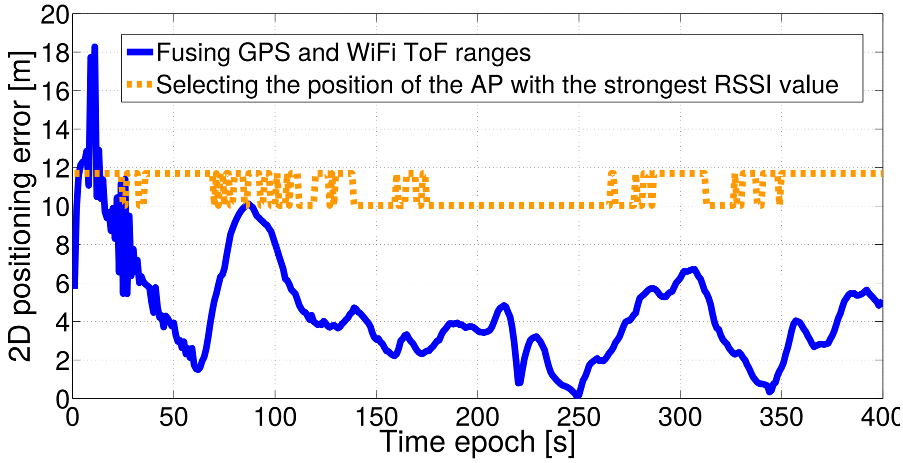


Figure 4.13: Constrained scenario with few ranges per technology.

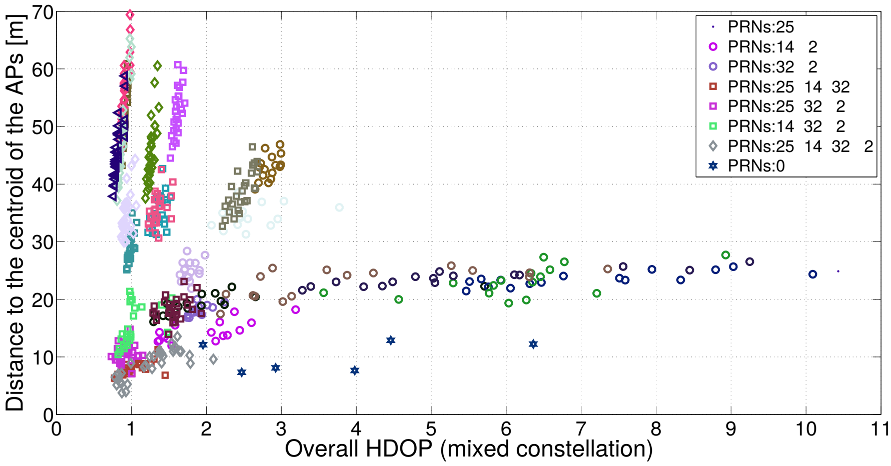


Figure 4.14: Distance to centroid versus HDOP pair for a specific set of satellites among all the ones in range. The color indicates a specific set of satellites and different values with the same color indicate a different set of APs (PRN (pseudorandom noise number) is a label used to uniquely identify a GPS satellite, ”PRNs:0” corresponds here to the pure WiFi case).

monolithic architecture. The RMSE (Root Mean Square Error) of fusing ranges is 3.90 m while selecting the AP with the strongest signal strength gives an error of 10.80 m.

4.9. Hybrid Positioning with Rich Number of Ranges

It is also fundamental to study scenarios where there is the availability of a rich set of GPS ranges and WiFi APs, which poses the problem of *which ranges to use* as input to the EKF. Our study is conducted using the main indoor scenario presented in Section 4.3, where some ranges can be largely affected by multipath.

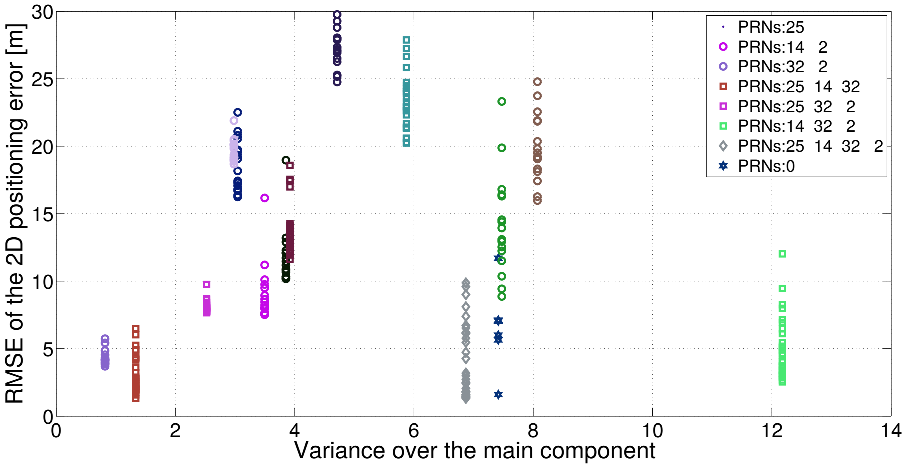


Figure 4.15: RMSE accuracy as a function of the variance over the principal component. (before clustering)

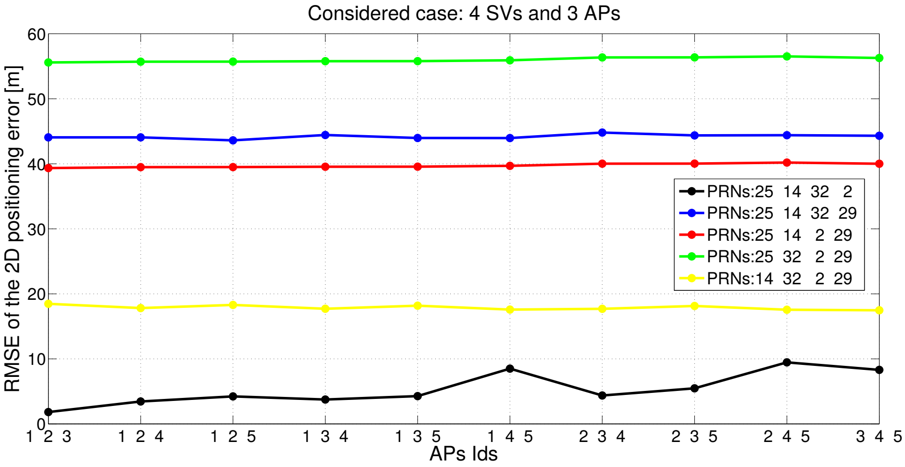


Figure 4.16: The RMSE accuracy (in meters) for bruteforce analysis using 4 satellites and 3 APs.

4.9.1. Geometrical-Statistical approach

First Observation.

In order to understand the problem space to select the set of anchors, we consider a client in a given position and in range of at least 5 anchors (GPS satellites and WiFi APs). Here, we perform a brute force analysis to understand what configuration of anchors would help getting an accurate position fix, measured with the root-mean-square error (RMSE) metric. For the setup of the scenario, we shortlist the satellites that are available all over a trace of 200 seconds. The accuracy is computed removing the transitory time to let the EKF filter converge. For an easy readability, we only show the specific case in Fig. 4.16 that summarizes the results for the case of 4 SVs and 3 APs. The figure shows that the choice of the GPS SVs (labeled using the PRN codes of each satellite, e.g. ”PRNs: 25 14 32 2”) affects much more the performance than the one of the WiFi APs (labeled as ”APs ids”).

Geometrical Metrics.

We propose to use two geometrical metrics for the selection of the anchors. First, we adopt the centroid of the APs as a reference point with respect to the estimated position of the client $\hat{\mathbf{p}}$. Second, the horizontal dilution of precision (HDOP) of all the anchors, we recall this metric that was previously introduced in Section 4.5.3.1 and provide further specific aspects. In GPS systems, the HDOP is a term that accounts for the multiplicative effect of the geometry of the satellites on the positioning accuracy [57].

$$dRMS = \sigma_{PR} \cdot HDOP \tag{4.16}$$

where σ_{PR} is computed as in eq. 4.11 and dRMS indicates the positioning error with 63% probability [57]. Smaller is the HDOP, smaller is the sensitivity of the position solution with respect to the errors in the ranges. We can use the same formulation for a multi-technology positioning system. In fact, the HDOP only depends on the unit vector of the direction between the receiver and the anchor rather than the coordinates and the distance separating them, that is, $(\mathbf{p}_{\text{anchor}} - \hat{\mathbf{p}})^T / ||\mathbf{p}_{\text{anchor}} - \hat{\mathbf{p}}||$, where $\mathbf{p}_{\text{anchor}}$ indicates the position of the generic anchor of the set $M = S\{I, J\}$. and $\hat{\mathbf{p}}$ indicates the estimated position of the client.

Let $\mathbf{P} = \begin{bmatrix} \mathbf{P_1} & \mathbf{P_2} & \dots & \mathbf{P_M} \end{bmatrix} \in \mathbb{R}^{3 \times M}$ be the matrix containing the ECEF (Earth-Centered Earth-Fixed) coordinates of these anchors and $\mathbf{p}$ the estimated position of the receiver at the epoch we want to compute the HDOP.

$$\mathbf{A} = \begin{bmatrix} (\mathbf{P_1} - \mathbf{p})^T / ||\mathbf{P_1} - \mathbf{p}|| & -1 \\ (\mathbf{P_2} - \mathbf{p})^T / ||\mathbf{P_n} - \mathbf{p}|| & -1 \\ \vdots & \vdots \\ (\mathbf{P_n} - \mathbf{p})^T / ||\mathbf{P_n} - \mathbf{p}|| & -1 \end{bmatrix} \tag{4.17}$$

We then formulate the matrix $\mathbf{Q} \in \mathbb{R}^{4 \times 4}$ as $\mathbf{Q} = (\mathbf{A}^T \cdot \mathbf{A})^{-1}$ and finally compute the HDOP as:

$$hdop = \sqrt{\mathbf{Q}_{11} + \mathbf{Q}_{22}} \tag{4.18}$$

We estimate the position in all the locations in the testbed using all possible set of anchors and plot the results in Fig. 4.14. The color of each result in the figure indicates the (Distance to Centroid, HDOP) pair for a specific set of SVs among all the ones in range. Different values with the same color indicate a different set of APs. We observe that the pairs (Distance to Centroid, HDOP) are clustered based on the set of SVs used for the position fix, alongside with all the combinations considering from one to the maximum number of WiFi APs.

Fig. 4.15 shows that based on the minimum variance over the principal component, the best cluster is the one that corresponds to ”PRNs: 32 2” (the actual PRNs are only shown for key cases in the figure to avoid heavy notations) while Fig. 4.18 tells that it is only a sub-optimal solution and that the optimal one is ”PRNs: 25 14 32”. The reason why this approach does not find the

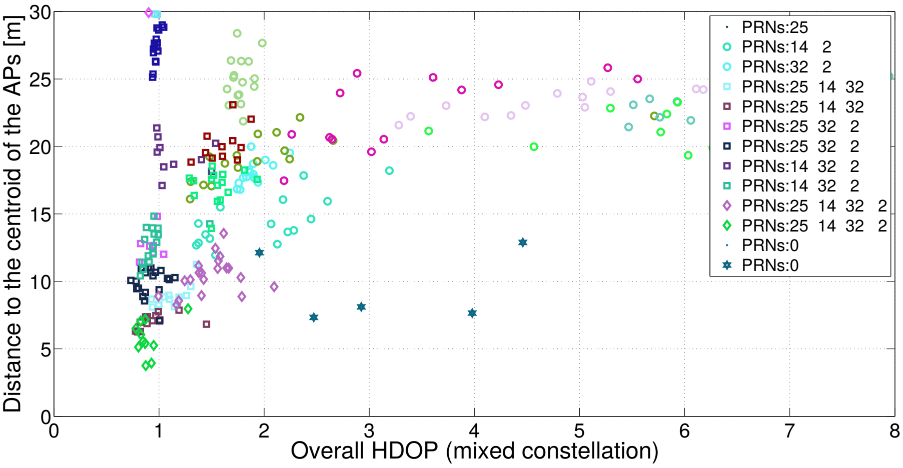


Figure 4.17: Distance to centroid versus HDOP pair after clustering

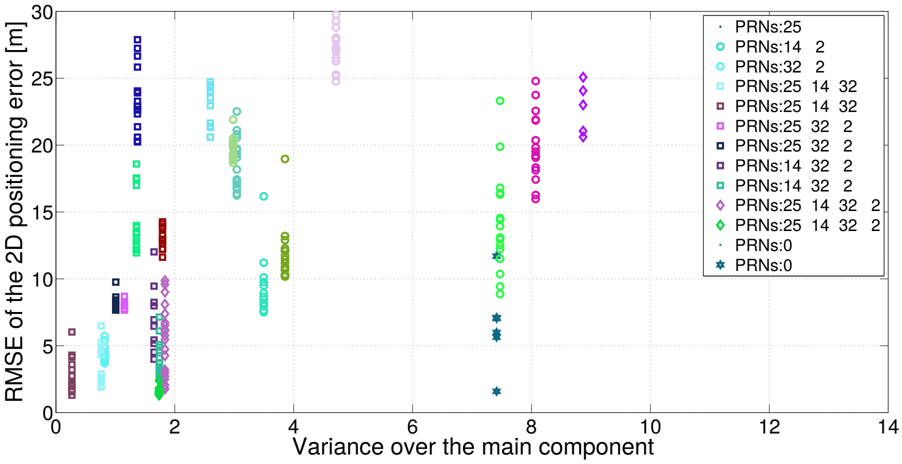


Figure 4.18: RMSE accuracy as a function of the variance over the principal component after clustering.

optimal solution is that the variance metric is sensible to the length of the data. The lengths of the clusters with few satellites are lower than the ones with a higher number of satellites since at least a total of 4 anchors is required for a position fix. To address this problem, we propose to perform a sub-clustering to the high-lengths clusters. In this way, we can apply the previous approach on clusters with relatively homogeneous sizes to alleviate the problem of variance sensitivity to the length of the data.

Subclutering with K-means clustering. We perform the k-means clustering on the sets of selected SVs with large sizes (corresponding to multiple options for the selection of APs). This allows us to operate with relatively homogeneous sizes. We then perform a principal component analysis (PCA) to compute the variance of the clustered data. Results are shown in Fig. 4.18. With this approach we are able to choose the best set of SVs ”PRNs: 25 14 32”. In fact, computing the RMSE of the position errors for a given selection of SVs, we observe that *less spread of the* *cluster tends to correspond to better accuracy*. The reason is that the geometry of the considered GPS SVs remains robust regardless of the set of APs they are fused with. In turn, this indicates that the *selected set of SVs* is relatively reliable. In the final step, *we select the APs* within the selected set of satellites such that the overall HDOP is minimal, which, as stated above, guarantees low sensitivity with respect to small variations in the noise in the ranges. The presented approach could run either on the mobile or could be offloaded to the cloud (that would receive GPS and WiFi range estimations).

4.9.2. The Geometrical-Statistical based Algorithm

Based on the study of the previous section, in Algorithm 1 we illustrate the pseudo-code. For clarity, the notation is given in what follows:

- Let $S\{I, J\}$ defined below be the mixed GPS-WiFi constellation formed by the set of satellites I and the one of APs J:

$$S\{I,J\} = \{\{sv_i\}_{i \in I}, \{ap_j\}_{j \in J} / I \subseteq \mathcal{SV}, J \subseteq \mathcal{AP}\}$$

- $\mathscr{C}_k(E)$ denotes the set of all k-combinations of the set E. $\mathscr{C}_0(E)$ simply corresponds to the empty set $\emptyset$.

- $\mathscr{C}_{[\![k,l]\!]}(E)$ denotes the set of all k-combinations to l-combinations of the set E.

- The cardinality of the previously defined set is computed as follows:

$$\#[\mathscr{C}_{[\![k,l]\!]}(E)] = \sum_{i=k}^{l} \binom{\#[E]}{i}$$

- $\lceil x \rceil$ denotes the ceiling of the real number x.

- n_{min} corresponds to the number of unknowns in the EKF state vector that are used for a position fix.

4.9.3. Evaluation of Anchor Selection

We perform an evaluation of our proposed solution for anchor selection. We conduct the evaluation with our system in phase II, using bias values of WiFi APs estimated in phase I (cf. Sec. 4.7.2). Fig. 4.19 shows the RMSE positioning error for different single and multi-technologies localization algorithms. On the left side we show a comparison of the different fusion strategies, while on the right we show a comparison of the fusion approach versus standalone WiFi with SNR measurements and standalone GPS estimator. For the SNR-based approach, we

Algorithm 1 Geometrical-Statistical Algorithm for Anchors Selection

```
Input: $\mathcal{SV}$, $\mathcal{AP}$
for $I \in \mathscr{C}_{[\![0,\#[\mathcal{SV}]]\!]}(\mathcal{SV})$ do
    Form $\mathcal{C}_I$, the set of all the mixed-constellations containing the set of satellites $I$ (for the pure WiFi case $I = \emptyset$) and all the possible combinations of WiFi APs of at least $n_{min}$ total anchors
    $\mathcal{C}_I$ is then the set of combinations of APs: $\mathscr{C}_{[\![max[1,(n_{min}-\#[\mathcal{SV}])],\#[\mathcal{AP}]]\!]}(\mathcal{AP})$
    for $J \in \mathscr{C}_{[\![max[1,(n_{min}-\#[\mathcal{SV}])],\#[\mathcal{AP}]]\!]}(\mathcal{AP})$ do
        Compute the position using the set $S\{I, J\}$,
        Compute its corresponding distance to the centroid of $\mathcal{AP}$
        Compute the mixed-constellation HDOP of $S\{I, J\}$: $hdop[S\{I, J\}]$
        Construct the matrix $M(\mathcal{C}_I)$ containing the previously computed metrics.
    end for
    Compare $\#[\mathcal{C}_I]$ to the pre-computed threshold $\eta$ based on combinatorial analysis: $k_I = \lceil \#[\mathcal{C}_I]/\eta \rceil$,
    Perform $k_I$-means clustering to sub-cluster $\mathcal{C}_I$ with respect to $M(\mathcal{C}_I)$ into $\{\mathcal{C}_I^1, \mathcal{C}_I^2, \dots, \mathcal{C}_I^{k_I}\}$.
    $\forall i \in [\![1, k_I]\!]$, apply the principal component analysis on $\mathcal{C}_I^i$ to compute its variance $\sigma_{I,i}^2$ over its main component.
end for
$\mathcal{C}_{\hat{I}}^{\hat{i}} = \underset{\substack{I \in \mathscr{C}_{[\![0,\#[\mathcal{SV}]]\!]}(\mathcal{SV}) \\ i \in [\![1,k_I]\!]}}{\operatorname{argmin}} \{\sigma_{I,i}^2\}_{I,i}$,
Find $\hat{J}$ that minimizes the $hdop$ such that $S\{\hat{I}, \hat{J}\} \in \mathcal{C}_{\hat{I}}^{\hat{i}}$ and compute the final position fix.
$S\{\hat{I}, \hat{J}\} = \underset{J \in \mathcal{C}_{\hat{I}}^{\hat{i}}}{\operatorname{argmin}} \{hdop[S\{\hat{I}, J\}]\}_{J \in \mathcal{C}_{\hat{I}}^{\hat{i}}}$
return $S\{\hat{I}, \hat{J}\}$
Compute the final position fix using the constellation $S\{\hat{I}, \hat{J}\}$
```

use the k-Nearest Neighbor machine learning algorithm (as typical in current WiFi-based location systems [52]).

Simply fusing pseudoranges from these SVs with all the available WiFi ranges is better than relying exclusively on GPS, with an overall accuracy of 11.78 m. As reference methodology, we use the RAIM algorithm that is designed for integrity monitoring [60] (cf. Appendix A.2). RAIM has the objective of removing one faulty range in case there is redundancy in the number of measurements. However, although some multipath can be detected by RAIM, it is known to fail to detect short delay multipath, which abounds in urban and indoor environments. This is confirmed by our study: trying to reject the non-reliable anchors using RAIM algorithm does not significantly enhance the positioning accuracy, with an average error of 11.07 m. Fig. 4.19 (left) shows that the statistical-geometrical selection outperforms any other fusion strategy. Using the anchors selection approach with clustering, we achieve an accuracy that is 74.2% higher than applying the anchors selection method without first sub-clustering the sets. From Fig. 4.19 (right), we can observe that our statistical-geometrical anchor selection algorithm drops down the average

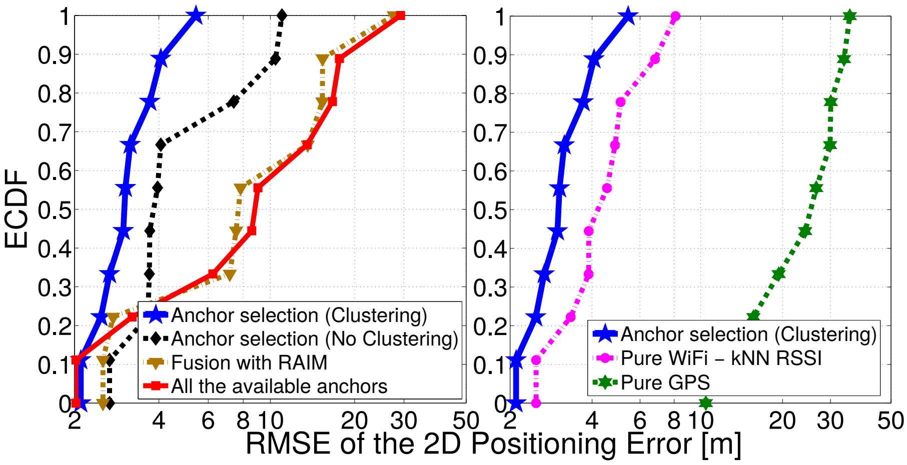


Figure 4.19: The ECDF of the RMSE (Root Mean Square Error) of the 2D positioning accuracy in meters across different *indoor* positions.

RMSE positioning error to 3.29 m, providing a gain of $8x$ with respect to standalone GPS, and better than standalone WiFi positioning with an average of 49.4% accuracy improvement. The latter would be the technology used by current monolithic architectures that do not exploit raw measurements and select the radio providing the best performing localization system.

4.10. Wrap-up & Summary

We proposed the idea that consists in complementing GPS pseudoranges measurements with terrestrial Time-of-Flight ranges to compensate the lack of visible SVs in indoor and dense urban environment. We introduced a two-phases fusion approach that estimates first the WiFi AP bias is introduced to greatly increase the autonomy of the system, the only assumption we kept is the prior knowledge of the APs positions' coordinates. Moving from technology selection to technology fusion approaches raise the problem of optimal selection of set of anchors in hybrid systems that we solved by introducing the Geometrical-Statistical method. In constrained environment we achieved a gain of 2.77x with respect to baseline approach, and in rich number of ranges scenario, gain of 8x versus pure GPS positioning and 50% versus pure WiFi.

Chapter 5

WiFi Proximity Metrics with Unknown Anchors Positions

5.1. Introduction & Motivation

In recent years, mobile smart devices have become ubiquitous, e.g., smart phones, personal health devices, smart watches, and other smart home appliances. Low cost requirements and integration with legacy networks lead to wide adoption of IEEE 802.11-based wireless communication (WiFi). Many such devices, and devices that are part of the so-called Internet-of-Things (IoT), interact with their physical environment and physically close communication partners either directly or via cloud platforms.

To determine the physical locations of such devices, WiFi-localization based approaches have become widespread. Typically, the mobile device collects a wireless fingerprint at their current location. This consists of set of MAC addresses of nearby Access Points (APs), and their received signal strength indicator called RSSI. This fingerprint data is sent to a third-party *cloud service* (such as Google, Apple, Skyhook, etc.), that estimates the device's location. The motivation for using a third-party cloud service is that fingerprint data is noisy, and it requires a large set of measurements from a large number of users and access to a large database of positions of APs in order to provide high-to-medium accuracy level.

In this work, we look at the problem of WiFi proximity metrics in positioning systems, i.e., *metrics that allow to estimate the spatial correlation or physical distance between two WiFi fingerprints*. Such metrics can be valuable in a number of scenarios:

- Measurement sanitization for user-assisted WiFi localization database building [33, 38];
- Fingerprint ambiguity estimation in an unknown environment [68];
- Detection of co-location for mobile applications on mobile devices (as attack on users' privacy, see [43] for Bluetooth).

There are three main challenges to address to effectively solve the problem of WiFi proximity metrics.

Challenge 1: Design of proximity metrics without using any external sensors. To the best of our knowledge, metrics that do not require any external sensors have received limited attention in related work. Yet, the availability of a solution in this space would allow a widespread adoption to any WiFi transceiver, from low-end IoT device to more powerful smartphone device.

Challenge 2: Metrics that can cope with probabilistic observations. Fingerprints might *not* contain information of all APs in range. There are two reasons:

- storage constraints might limit the number of APs stored for each fingerprint;
- collisions, time-out, and channel hopping during WiFi scanning can prevent deterministic observations of nearby APs.

Storage of WiFi fingerprints can require relatively large amount of memory, if the MAC address (6 Bytes) and an RSSI value (1 Byte) is stored for each AP. Given the growing density of WiFi APs, single scans for neighboring APs can easily return 30 and more results. Storing an unlimited number of APs per fingerprint can thus violate memory constraints.

Challenge 3: Metrics that do not have access to large databases of known locations of APs. In this work, we are interested in metrics that do not require prior knowledge of the locations of APs in the environment. In particular, large-scale records of AP locations are proprietary data owned by few companies, and not accessible by the public.

Finally, related work lacks of *large scale real-world data evaluation of proximity metrics*, which are needed to assess the soundness of the proposed solution.

Our main contributions are listed in what follows:

- We perform an extensive evaluation of a novel Jaccard Index-based metric and two prior work metrics over two main datasets: i) an artificial dataset based on simplified propagation models and perfect knowledge of the true locations, and ii) a large-scale, real-world WiFi fingerprint data set consisting of $200,000$ fingerprints resulting from a large deployment of wearable WiFi sensors [76].
- We identify key drawbacks of all three metrics using our real-world dataset, analyze the causes and propose a model to explain the issue. Further analysis of our dataset confirms that our model closely matches observed data.
- Based on our error model, we propose three improved distance metric definitions. Our proposed metrics only require two WiFi fingerprints readings, and enable mobile devices to compute the results i) without continuous requests to the third-party cloud service, ii) without disclosing the location to the cloud service and to neighbor nodes, and iii) with limited requirements of local storage and low computational and implementation complexity.

This work is structured as follows. In Section 5.2, we introduce the system model, metrics from related work and the datasets used (real and simulated data). In Section 5.4.1, we introduce the detailed problem, our first attempts at addressing it, and investigations into mismatch between performance on simulated data versus real data. To address performance drop with real data, we introduce improved metrics Section 5.4.3 and show that they perform well in both settings. We conclude in Section 5.5.

5.2. Background

5.2.1. System Model and WiFi Fingerprints

In this work, we focus on IEEE 802.11b (which we will refer to as WiFi), but general principles hold for other wireless standards as well. We consider a node with wireless transceiver that is capable of observing the presence of WiFi APs and their RSSI values. To simplify the discussion, we assume that the infrastructure is static, and the node is moving over time. We also assume that omnidirectional antennas are used. The density of APs in range can vary from 0 to more than 40.

The node gathers measurements of WiFi fingerprints over time. Then, the node uses the proximity metric to find an estimate of the moved distance for consecutive fingerprints. We use the following notation in this work to refer to WiFi fingerprints. We define n as our node of interest, a set $\mathbb{A}$ of all APs in the target area, and $\mathbb{N} \subset \mathbb{A}$ as the subset of nearby APs within $\mathbb{A}$ that are in range of node n. We use $|\mathbb{N}|$ to indicate the cardinality of a set, i.e. the count of distinct items in the set. For integers $|x|$ is the absolute value of x.

Let $o_i = (\text{RSSI}_i, \text{MAC}_i)$ denote the AP observation of n for AP i. If i is not in $\mathbb{N}$, then $\text{RSSI}_i = 0$. Then, $F = \{o_i\}, \forall i \in \mathbb{N}$ is a WiFi fingerprint which corresponds to the list of observations of currently neighboring WiFi APs. In case of multiple fingerprints, we will use a notation of F_a with $\mathbb{N}_a$ as set of neighbors.

5.2.2. Related Work Metrics

In the following, we use $\mathbb{N}_a, \mathbb{N}_b$ as set of neighboring APs of fingerprint F_a, F_b, respectively. The observation o_{ai} refers to AP $i \in \mathbb{N}_a$ in fingerprint F_a. We consider the following two metrics, MetricE and MetricM, from related work. We present them only for the sake of comparing them to the ones we design in this work.

MetricE (Euclidean Distance): In [8], the authors discuss RSSI proximity metrics. They propose to compute the Euclidean distance between the RSSI vectors $\{\text{RSSI}_{ai}\}_{i \in \mathbb{N}_a \cap \mathbb{N}_b}$ and $\{\text{RSSI}_{bi}\}_{i \in \mathbb{N}_a \cap \mathbb{N}_b}$ from the set of APs $\mathbb{N}_a \cap \mathbb{N}_b$ present in both fingerprints F_a and F_b:

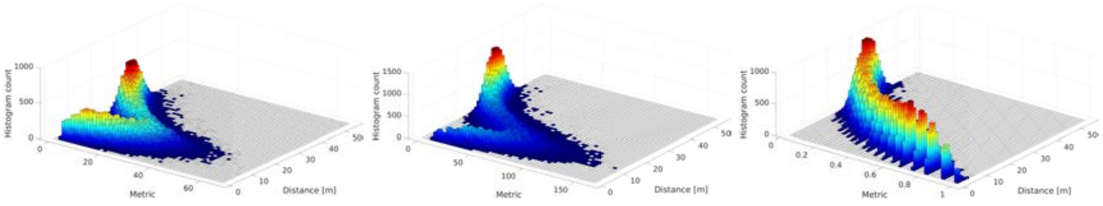
$$m(F_a, F_b) = \sqrt{\sum_{i \in \mathbb{N}_a \cap \mathbb{N}_b} \left[\text{RSSI}_{ai} - \text{RSSI}_{bi}\right]^2} \tag{5.1}$$

MetricM (The ”Manhattan” distance): In [8], the authors also propose to use the Manhattan distance that refers to the sum of the absolute differences instead of the Euclidean one.

$$m(F_a, F_b) = \sum_{i \in \mathbb{N}_a \cap \mathbb{N}_b} |\mathrm{RSSI}_{ai} - \mathrm{RSSI}_{bi}| \tag{5.2}$$

5.2.3. NSE Dataset

Our evaluation uses a large-scale real-world dataset, collected as part of the National Science Experiment (NSE) project in Singapore. We now briefly introduce the SENSg devices that are used to gather the dataset.



(a) MetricE (b) MetricM (c) MetricJ

Figure 5.1: Simulation-based evaluation of initially proposed vs prior work metrics.

SENSg sensors: A total of 50,000 devices are produced and used by students at schools. The students wear the devices for a week or longer, and collect data about their daily life. The data is automatically uploaded to a cloud platform, and made available to the students to analyze. The devices are called SENSg [76], and they record WiFi fingerprints (as defined in this work) every 12 seconds. Using a third party API, the WiFi fingerprints are mapped to location estimates after the data is uploaded to the cloud. The SENSg devices store only the 20 APs with highest RSSI per fingerprint. Storing the MAC address and received signal strength for each AP requires 7 Bytes, so a fingerprint with 20 observed APs is 140 Byte large.

Measurements and data gathering: The real world datasets used in this work are gathered by students during the NSE project. In particular, the dataset used in this work is a subset of all fingerprints taken. For all fingerprints used, we also have a location estimate by the third party API. Location provided by the third party API is subject to accuracy errors, as in typical cloud-based location based-systems.

5.2.4. Simulation setup

In addition to the real world dataset, we generate an artificial dataset with $200,000$ fingerprints as in the real measurement data. The fingerprints are randomly distributed in an area which is roughly equivalent to the area covered in the real-world dataset. We generate this second dataset to have a better control over noise and other factors that lead to unexpected behavior in the observed

APs, and their RSSI values. In addition, this allows us to know the true location of fingerprints, which will be exploited in the evaluation.

To simulate path loss $L(d)$ and the resulting RSSI, we use the 802.11 propagation model E presented in [21] that takes into account LOS (Line-of-Sight) and NLOS (Non-Line-of-Sight) channels, breaking point distance in which the attenuation slope drastically changes because of the scattering effect, log-normal shadowing, fading and other physical channel phenomena. $L(d)$ consists in the sum of a freespace component $L_{FS}(d)$ whose slope depends on the distance d in comparison to the aforementioned breakpoint distance d_{BP} and a shadow fading component $S_F(d)$ that accounts for the large scale scattering.

$$L(d) = \begin{cases} L_{FS}(d) + S_F(d) & d \leq d_{BP} \\ L_{FS}(d_{BP}) + 10\alpha_2 \log_{10}\left(\frac{d}{d_{BP}}\right) + S_F(d) & d > d_{BP} \end{cases} \tag{5.3}$$

α_2 is the attenuation slope after d_{BP} while $L_{FS}(d)$ is computed with a lower attenuation slope α_1 before d_{BP}.

$$L_{FS}(d) = 10\alpha_1 \log_{10}\left(\frac{4\pi f d}{\mathfrak{c}}\right) \tag{5.4}$$

f denotes the frequency and $\mathfrak{c}$ the speed of light in vacuum. The shadow fading component is modeled by a log-normal distribution centered in zero with a standard deviation σ_{SF}. The constants for model E as introduced in [61] are the following: $d_{BP} = 20$ meters, $\alpha_1 = 2$, $\alpha_2 = 3.5$ and $\sigma_{SF} = \{3, 6\}$.

5.3. NSE Experiment and Artificial dataset

5.3.1. SENSg sensors

A total of 50,000 devices are produced and used by students at schools. The students wear the devices for a week or longer, and collect data about their daily life. The data is automatically uploaded to a cloud platform, and made available to the students to analyze. The devices are called SENSg [76], and they record WiFi fingerprints (as defined in this work) every 12 seconds. Using a third party API, the WiFi fingerprints are mapped to location estimates after the data is uploaded to the cloud. The SENSg devices store only the 20 APs with highest RSSI per fingerprint. Storing the MAC address and received signal strength for each AP requires 7 Bytes, so a fingerprint with 20 observed APs is 140 Byte large.

5.3.2. Measurements and data gathering

The real world datasets used in this work are gathered by students during the NSE project. In particular, the dataset used in this work is a subset of all fingerprints taken. For all fingerprints used, we also have a location estimate by the third party API. Location provided by the third party API is subject to accuracy errors, as in typical cloud-based location based-systems.

5.3.3. Simulation setup

In addition to the real world dataset, we generate an artificial dataset with $200,000$ fingerprints as in the real measurement data. The fingerprints are randomly distributed in an area which is roughly equivalent to the area covered in the real-world dataset. We generate this second dataset to have a better control over noise and other factors that lead to unexpected behavior in the observed APs, and their RSSI values. In addition, this allows us to know the true location of fingerprints, which will be exploited in the evaluation.

To simulate path loss $L(d)$ and the resulting RSSI, we use the 802.11 propagation model E presented in [21] that takes into account LOS (Line-of-Sight) and NLOS (Non-Line-of-Sight) channels, breaking point distance in which the attenuation slope drastically changes because of the scattering effect, log-normal shadowing, fading and other physical channel phenomena. $L(d)$ consists in the sum of a freespace component $L_{FS}(d)$ whose slope depends on the distance d in comparison to the aforementioned breakpoint distance d_{BP} and a shadow fading component $S_F(d)$ that accounts for the large scale scattering.

5.4. Proximity metrics for WiFi Fingerprints

5.4.1. Proximity Metrics for WiFi Fingerprints

In this section, we summarize our problem statement, and then present a number of candidate proximity metrics that will be evaluated later.

5.4.1.1. Problem Statement

Our goal is to provide a metric $m(F_a, F_b)$ able to estimate the expected spatial correlation between two fingerprints. The metric will be optimized for accuracy in the estimate, and low computational cost. Intuitively, that metric should be 1 if two fingerprints are taken at the exact same location, and 0 if they are completely uncorrelated (e.g., no single access point was observed by both fingerprints).

Consequently, we define the proximity metric as a function $m(F_a, F_b) = y$ between the fingerprints F_a and F_b, with $0 \leq y \leq 1$.

5.4.1.2. Problems with Metrics from Related Work

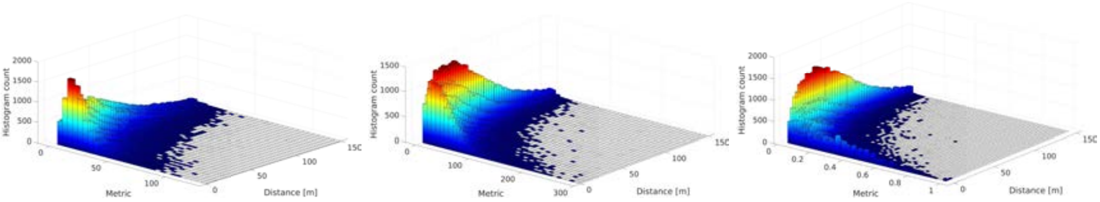
We evaluate MetricE and MetricM from related work over our simulated data set. We summarize the results in Figure 5.1 (a) & (b). From our study, we find that both have several issues: i) they are not normalized, but return a value ≥ 0, with smaller values indicating proximity; ii) they do not work well for distances larger than 15 meters, for which only few mutual APs are observed. We note that the original work used the metrics to find closest matching fingerprint pairs, and not to estimate exact distances. As it can be seen in Figure 5.1, for distances greater than 15 meters, the metric score is decreasing on average, leading to values indicating closer proximity. One explanation for that behavior could be that with increasing distance between the fingerprint locations, fewer mutual APs are observed and the sum of RSSI differences will decrease with increasing distance. This is a practical issue, as low metric score could either indicate close proximity, or distances of more than 15 meters. To the best of our knowledge, such issues were not discussed before in related work.

5.4.1.3. Using Jaccard-Index to Reward Mutual Observations

Based on the above findings, intuitively, a metric should score high if a large fraction of the APs observed in two fingerprints are shared, i.e. mutually observed. Based on that, our initial proposal is to use a metric that contains a factor relating to the Jaccard Index [47], defined as:

$$m(F_a, F_b) = \frac{|\mathbb{N}_a \cap \mathbb{N}_b|}{|\mathbb{N}_a \cup \mathbb{N}_b|} \tag{5.5}$$

In other terms, for two fingerprints, the Jaccard Index is a ratio of number of mutually observed APs, divided by the total number of observed APs. To simplify the discussion, our first proposed metric MetricJ is exactly the Jaccard Index, see Eq. 5.5.



(a) MetricE (b) MetricM (c) MetricJ

Figure 5.2: Real data-based evaluation of initially proposed vs prior work metrics.

We note that MetricJ is range-free, i.e. RSSI values are not directly used, while MetricE and MetricM measure the similarity of RSSI values of APs that are observed in both fingerprints. Nevertheless, we evaluate the use of MetricJ as distance metric, and discuss the results. We later discuss the use of the Jaccard Index as factor to scale other range-based metrics.

We evaluate the performance of MetricJ compared to MetricE and MetricM with two sets of data: simulated fingerprints, and a large set of real-world data (see Section 5.2).

5.4.1.4. Simulation-based Evaluation

We start with a simulation-based evaluation, using the dataset described in Section 5.3.3. To evaluate the quality of the metric, we compare the metric score to the true distance between the

Metric	Dataset	Correlation	Improv. Correlation
MetricJ	Artificial	0.91	0.91
MetricE	Artificial	0.59	0.94
MetricM	Artificial	0.72	0.94
MetricJ	Real-world	0.46	0.49
MetricE	Real-world	0.31	0.53
MetricM	Real-world	0.41	0.54

Table 5.1: Summary of Spearman correlation values of original and improved metrics on our datasets.

locations at which the fingerprints are taken. We compute the Spearman correlation [17] between the two values to obtain a quantitative result. Unlike the widely used Pearson correlation, the Spearman correlation evaluates the monotonic relationship and it is then a better fit for non-linear correlation studies. In addition, the Spearman correlation is proven to be robust in the sense of being resistant to outliers [17].

From visual inspection of our simulation results presented in Figure 5.1, MetricJ is superior to the two related work metrics. This is confirmed by the correlation scores of 0.59 (MetricE), 0.72 (MetricM) and 0.91 (MetricJ) (see Table 5.1).

5.4.1.5. Real-World Data-based Evaluation

For this dataset (introduced in Section 5.2.3), we do not have accurate ground truth locations available. Instead, we will use the distance between the fingerprints based on the third party's localization result (which itself is a noisy estimate) for the performance evaluation. We evaluate around $200,000$ fingerprints located all around Singapore.

As it can be observed in Figure 5.2, all three metrics perform much worse than expected on the real dataset. We note that for MetricJ, a value of 1 should correspond to a small distance, while for the other two metrics, smaller values mean shorter distances. For all three metrics, distances of $<$10m generally do have expected metric scores, but the Spearman correlation between distance and score is not very strong: 0.46 (MetricJ), 0.31 (MetricE), 0.41 (MetricM). While MetricJ's correlation score is still better than the other two metrics, MetricJ has very low values even for short distances, and high scores are almost never reached. Clearly, the performance predicted based on simulations is not achieved when the evaluation is done using our real-world dataset. Surprised by those results, we set out to investigate the cause, and possible mitigations.

5.4.2. Analysis of MetricJ

We now present our analysis of the causes for the bad performance of MetricJ on the real data set. We start by analyzing metric scores for fingerprints that are estimated to be taken at the same locations. We find that MetricJ scores are low because the actual number of mutually

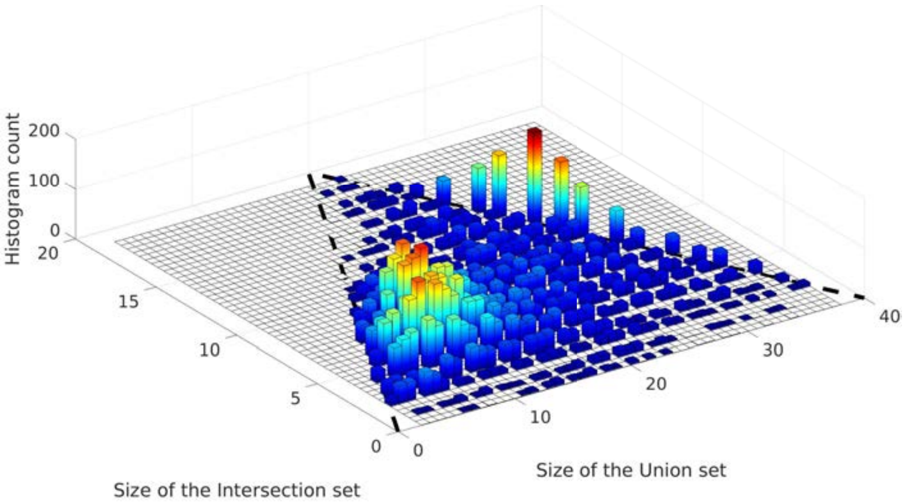


Figure 5.3: Jaccard index components: the number of mutually observed APs versus the total number of APs observed in two fingerprints that are at most 1 meter distance apart.

observed APs is much smaller than expected compared to the total number of APs observed in both fingerprints.

To confirm that finding, we select $11,433$ fingerprint tuples that are within respective estimated distance according to the third-party cloud service of 1 meter or less. We then compute the numerator and divisor of the Jaccard index and show the results in Figure 5.3. Points on the dashed diagonal lines indicate that both fingerprints contain the same set of observed APs. We expected to see the same or similar set of APs in both fingerprints, with diagonal up to a divisor score of 20 (c.f. Section 5.2.3), and only few cases of divisor scores higher than 20. Instead, the results show that there are *two distinct clusters*: around (7,12) and (14,28). Given the construction of MetricJ, those clusters would lead to metric scores of around 0.5, although scores of 1 would be expected given the short distance between the fingerprints. The question is now: *why there are so few APs mutually observed for fingerprints taken at the same location?*

5.4.2.1. Probabilistic Observations of APs

Our hypothesis is that the fingerprint collection process on the devices suffers from a probability e to miss a nearby AP completely (in addition to expected RSSI variations). More formally, e would lead to the following expected ratio of mutually observed APs versus total APs in the shared neighborhood:

(a) Probability of mutual observation: $P_m = (1 - e)^2$.
(b) Probability of having at least one observation: $P_o = 1 - e^2$.

This results in the following maximal Jaccard Index MJI for n APs in the shared neighborhood:

$$\text{MJI} = \frac{P_m * n}{P_o * n} = \frac{P_m}{P_o} = \frac{1-e}{1+e}$$

In other words, an effective upper bound for our metric score depends on e, even if fingerprints are taken from the same location.

5.4.2.2. Probability e of missing a nearby AP

First cluster. As expressed above, we have identified two clusters. Our guess is that the cluster around (14,28) is likely due to the limited cache of 20 best APs per fingerprint.

Second cluster. For the cluster around (7,12), we speculate that this is due to the limited scanning capabilities of WiFi chipsets. WiFi nodes scan each WiFi channel for APs for a limited time (e.g. 120-180 ms). As WiFi APs typically transmit beacons every 100 ms, channel congestions may cause that beacons transmitted during the time spent on that channel are lost due to the likelihood of beacons collisions and hidden nodes. We leave a more detailed investigation for future work.

Based on the clusterization above, we analyze the real-world dataset to attempt to give an estimated value of e due to caching and channel congestion. We compare the number of APs within a range of 20 meters from two fingerprints separated by a maximum distance of 1 meter to the total number of APs present in both fingerprints. The 20 meters radius we choose corresponds to the breakpoint distance after which the signal is more likely to suffer from an attenuation of a higher slope factor due to obstacles.

$$e = \frac{\mathbb{E}(\text{\# of APs in FPs of d} \leq 1\text{m})}{\mathbb{E}(\text{\# of APs within 20m})} = \frac{17.16}{67.62} = 0.25 \tag{5.6}$$

5.4.3. Improved Metrics

5.4.3.1. Candidate Metrics

We now present a set of candidate metrics to improve on MetricJ with regards to imperfect observations of APs, and to leverage RSSI values. In general, the metric should incorporate two properties: a) a factor reflecting on similarity in observed APs in both fingerprints, and b) counterbalance as much as possible the effect of the previously computed probability e.

We now explore options for both a) and b), leveraging our insights as presented in Section 5.4.4. In particular, we will convert the a)-related components of MetricE and MetricM into factors that range from 0 (for no similarity) to 1 (for maximal similarity) by imposing a limit on the RSSI difference for mutually observed APs, and a scaling factor s. In addition, we will introduce the same b)-related factor for all three metrics discussed.

Similarity score for mutually observed APs: In MetricE and MetricM, differences of RSSI values for mutually observed APs are computed, of which the squared or absolute value is then summed up. The resulting values are in practice between 0 and 250 (see Figure 5.2). We now replace that with a construction that returns values between 0 and 1 for each mutually observed AP. Together with the normalization factor, that will lead to an overall metric score between 0 and 1 (where larger scores indicate higher spatial correlation). We define that similarity score for two fingerprints F_a, F_b as follows:

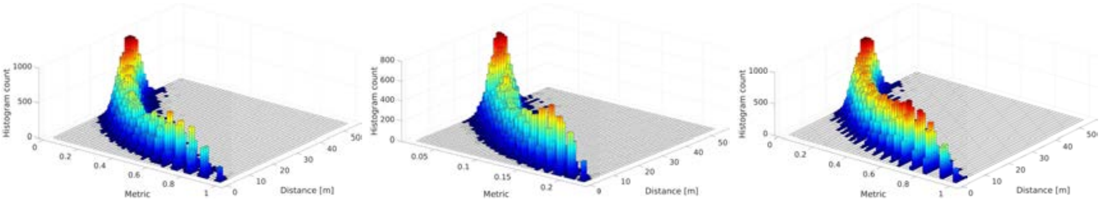
$$\delta(o_{ai}, o_{bi}) = 1 - \frac{|\text{RSSI}_{ai} - \text{RSSI}_{bi}|}{max(|\text{RSSI}_{ai}|, |\text{RSSI}_{bi}|)} \tag{5.7}$$

Counterbalance the effect of the probability e: In MetricJ, normalization (through the divisor) was based on the size of the union set of observed APs. As discussed in Section 5.4.2, the observed value of that size (and size of the intersection set) is biased by e. The idea we adopt limits the number of considered mutually observed APs for metrics computations. As the main peak in Fig. 5.3 is around (7,12), we propose to set a maximum number of APs, $\#^{\text{APs}}_{\text{max}} = 7$ over which the similarity score defined in Eq. 5.7 is computed, rather than the whole set $\mathbb{N}_a \cap \mathbb{N}_b$. These $\#^{\text{APs}}_{\text{max}}$ APs are chosen as those in $\mathbb{N}_a \cap \mathbb{N}_b$ with the lowest absolute difference of corresponding RSSIs between both FPs (Fingerprints) $|\text{RSSI}_{ai} - \text{RSSI}_{bi}|$, we denote this set as: $\mathcal{L}_{a,b}(\#^{\text{APs}}_{\text{max}})$

MetricJ+i (improved MetricJ): In this metric, we extend MetricJ with the upper bound on the intersection size to compensate e as discussed above. The resulting metric is:

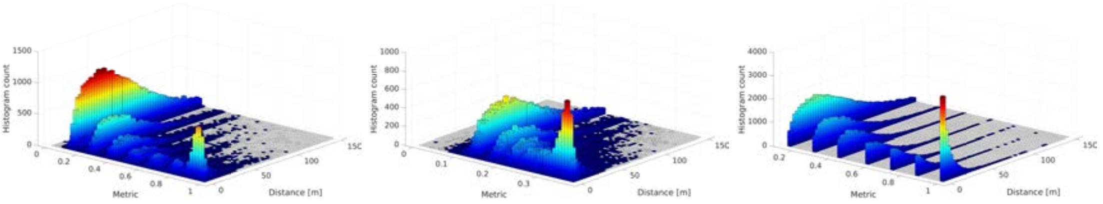
$$m(F_a, F_b) = \frac{\sum_{i \in \mathcal{L}_{a,b}(\#^{\text{APs}}_{\text{max}})} 1}{\#^{\text{APs}}_{\text{max}}} \tag{5.8}$$

Here, we choose to represent $|\mathcal{L}_{a,b}(\#^{\text{APs}}_{\text{max}})|$ as $\sum_{i \in \mathcal{L}_{a,b}(\#^{\text{APs}}_{\text{max}})} 1$ to highlight similarities to the other two improved metrics.



(a) MetricM+i (b) MetricE+i (c) MetricJ

Figure 5.4: Artificial data-based evaluation of improved metrics.



(a) MetricM+i (b) MetricE+i (c) MetricJ+i

Figure 5.5: Real data-based evaluation of improved metrics

MetricM+i (improved MetricM): The next metric is the Manhattan distance-based MetricM,

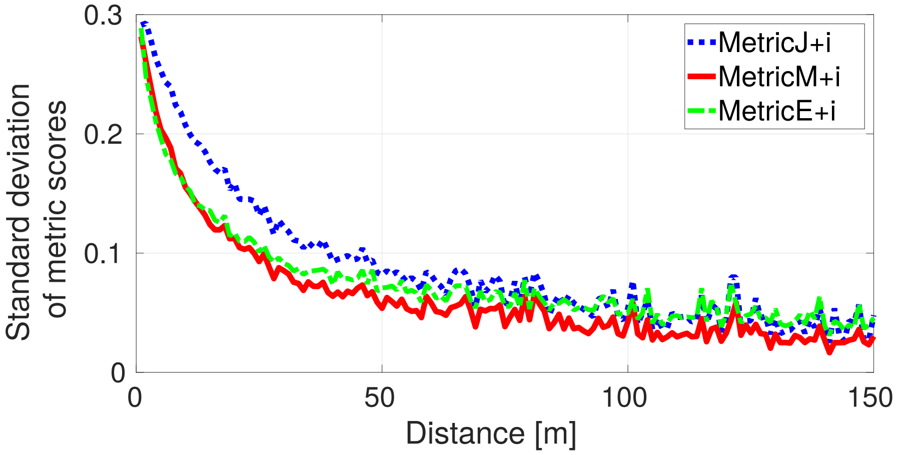


Figure 5.6: The standard deviation of metrics scores computed for FPs separated by distances ranging from 1 to 150 m (real measurement dataset).

improved with our new similarity function $\delta(F_a, F_b)$ and the compensation for the effect of the probability e.

$$m(F_a, F_b) = \frac{\sum_{i \in \mathcal{L}_{a,b}(\#^{\text{APs}}_{\text{max}})} \delta(o_{ai}, o_{bi})}{\#^{\text{APs}}_{\text{max}}} \tag{5.9}$$

MetricE+i (improved MetricE): The last metric is MetricE with our new similarity function $\delta(F_a, F_b)$ and the compensation for the effect of the probability e.

$$m(F_a, F_b) = \frac{\sqrt{\sum_{i \in \mathcal{L}_{a,b}(\#^{\text{APs}}_{\text{max}})} \left[\delta(o_{ai}, o_{bi})\right]^2}}{\#^{\text{APs}}_{\text{max}}} \tag{5.10}$$

5.4.4. Evaluation

We evaluate the metrics with simulated and real data, with results shown in Figure 5.5. The Spearman correlation scores are as follows: 0.49 (MetricJ+i), 0.54 (MetricM+i), and 0.53 (MetricE+i). All correlation results are summarized in Table 5.1. We conclude that both proposed improvements (similarity score and e-effect compensation) together improve the previously discussed metrics, with all three metrics performing similarly in terms of Spearman correlation.

The Spearman correlation is a good indicator of metrics performance. However, it does not capture other aspects necessary to compare the proposed metrics, such as the *robustness*. We introduce robustness as a performance measure in the sense that for a specific physical distance between FPs, the corresponding metric score should be as consistent as possible across different conditions and scenarios. For the real measurement dataset, we show in Fig. 5.6 the standard deviation of metrics scores vectors of size 3000 for each distance from 1 to 150 meters with a ± 0.1 m margin. Naturally, the lower this standard deviation is, the more robust is the corresponding metric. Although the 3 metrics compared in this Section show more or less similar Spearman correlation coefficients, Fig. 5.6 demonstrates that MetricM+i is the most robust one compared to MetricE+i and MetricJ+i.

5.5. Wrap-up & Summary

In this work, we discussed proximity metrics for WiFi fingerprints that do not need external sensors and do not have access to the locations of APs. Using real data from a large dataset as well as simulated data, we have shown that metrics proposed in related work do not perform as expected in noisy real datasets, and we have proposed a range of alternatives. We have also shown that access points might be missed in real environments with some probability e and proposed an upper bound for metric scores as a function of e. Based on those insights, we have improved our proposed metrics. The best performing metric (MetricM+i) has resulted in a Spearman correlation score of 0.54 with the real dataset and 0.94 with the artificial one.

Chapter 6

Conclusion and Future Work

We have proposed to fuse timing (Time-of-Flight (ToF)) range measurements of GPS and WiFi technologies, in order to circumvent the limitations of the individual radio access technologies and improve the overall localization accuracy and pervasiveness in several contexts. We first investigated WiFi echo technique, sources of noises affecting WiFi ToF measurements, and proposed two different shortest path estimators, one that requires environment training, and one that alleviates this need towards more autonomous systems without a substantial loss of accuracy with respect to the first. We have shown that fusing ranges is beneficial both in the scenarios where the standalone technology does not have enough anchors to provide a position fix and in the scenarios where there exists a rich set of multiple ranges. For the latter case, we have found that the intelligent selection of the subset of ranges with accurate timing information is the key to achieve the best positioning accuracy. We have introduced a novel geometrical-statistical approach to select only the ranges that could provide high accuracy and low horizontal dilution of precision (HDOP). In each scenario, a two-phase EKF algorithm is responsible to provide the position fix. In constrained environments, our system can provide a position fix in scenarios where single technologies with limited anchors in range need to rely on simple metrics. In an indoor scenario with a rich set of visible ranges, yet with GPS and WiFi measurements affected by multipath errors, our results show that our system can achieve a performance gain of 8x versus pure GPS positioning and 50% versus pure WiFi. Our approach is a step beyond classical approaches where technologies in multi-radio access are considered as monolithic entities. It can be extended to other communication technologies that provide timing information such as cellular networks. Eliminating the assumption of prior knowledge availability of WiFi APs positions' coordinates as well as the computational overhead of the solution we propose with a rich number of ranges that needs further investigation are the plan for our future work. Additionally, we came up with the design of proximity metrics for WiFi fingerprints that do not need external sensors and do not have access to the locations of APs. The best performing metric we propose has resulted in a Spearman correlation score of 0.54 with the noisy real dataset considered for the evaluation, and 0.94 with the artificial one.

Appendix A

Appendix

A.1. The GPS position simulator based on real traces

We collect real traces using the Evaluation Kit with Precision Timing manufactured by U-blox and equipped with an active GPS antenna of type u-blox ANN-MS, connected by a 5 meters coaxial cable. This device runs the U-center software for configuring the GPS receiver's settings, and testing, visualizing and analyzing the collected data. We choose this device because it presents the advantage of storing the raw data that we need to evaluate the positioning algorithms and deeply analyze the sources of error, these logs are stored as a single .ubx file.

A.1.1. The RINEX format

Since the GPS log data is stored in a proprietary file format .ubx proper to the u-blox receiver, we convert it to a more manufacturer independent file format. We opt for the Receiver Independent Exchange Format (RINEX), a raw navigation system data format that does not depend on the receiver's manufacturer or its special features. The data logs are classified in different files; the ones of interest for this study are the observation and navigation files (.obs and .nav).

A.1.1.1. Observation file

This file contains the pseudoranges, the code phase, the Doppler shift and the signal strength measurements from every single visible satellite mapped by their pseudorandom noise numbers (PRN) at every GPS epoch when the measurements were carried out.

A.1.1.2. Navigation file

Additionally to the ephemeris data broadcasted by the satellites that includes the SV clock bias, drift and drift rate, the Keplerian parameters, perturbation parameters and additional information proper to each SV, this file contains the ionospheric correction parameters and the terms of polynomial that are necessary to compute the satellite clock offset.

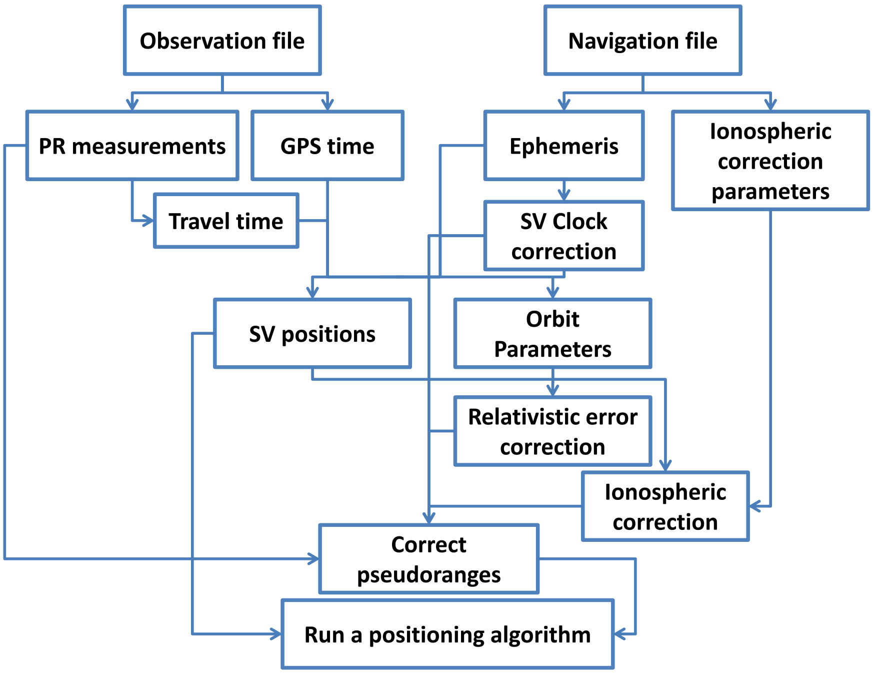


Figure A.1: Flowchart of our GPS simulator based on real traces

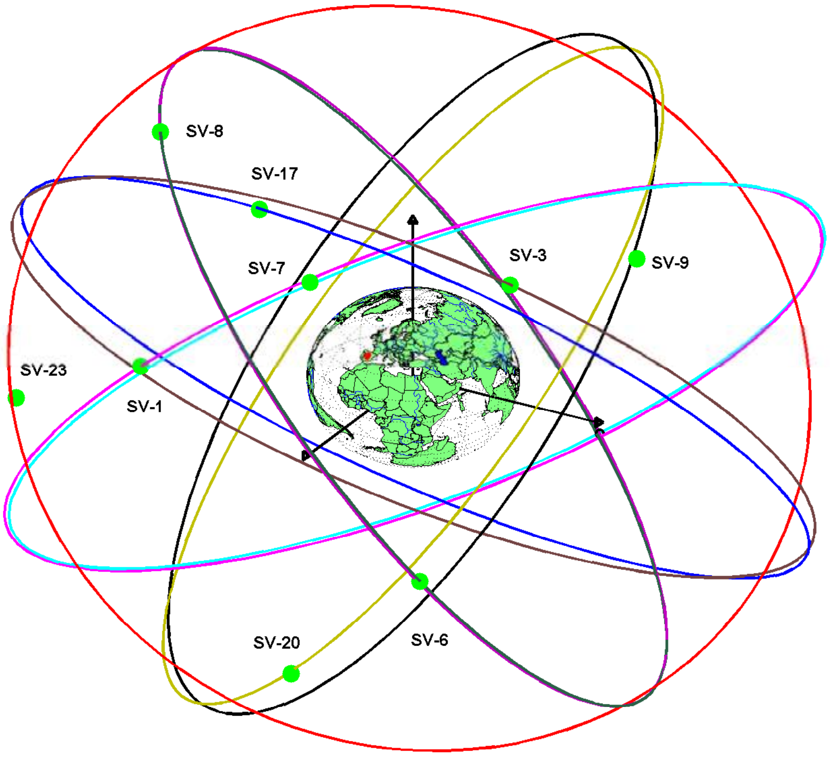


Figure A.2: Graphical representation of the GPS position, satellite and their orbits.

To convert the data logs from .ubx to RINEX we use RTKLIB, an open source program package for GPS positioning.

A.1.2. The MATLAB implementation of the Simulator

We implement a GPS simulator in Matlab that starts from the raw data logs in a RINEX format to perform all the functions related to the satellites' positions computation and the pseudoranges corrections introduced in Section 4.4. The flowchart of the simulator is depicted in Fig. A.1 and a graphical representation of the GPS position, satellite and their orbits in Fig. A.2. The simulator allows us to run the positioning algorithm we want to evaluate for the receiver's position calculation at every epoch.

A.2. RAIM for Multi-Technology Positioning

As reference methodology, we use the RAIM algorithm that is designed for integrity monitoring. RAIM is used for applications such as aviation and naval navigation for the case of pure GPS localization.

However, RAIM is designed for integrity monitoring and specially for aviation applications, where the prevalent source of errors are quite different to those in urban environments. Although some multipath can be detected by RAIM, it fails for detecting short delay multipath, which abounds in urban and indoor environments. However, it can be used to detect strong multipath components. The approach described in [60] takes benefit of the redundancy provided by the availability of more satellite signal measurements than the minimum required for the system to be capable to compute the receiver's position. This minimum corresponds to 4 (receiver's 3D coordinates (x, y, z) + the receiver's clock bias b_{GPS}). The linearized measurement matrix $\mathbf{H}$ undergoes a QR decomposition to derive the parity vector $\mathbf{p}$ (from a submatrix $\mathbf{P}$ of this decomposition and the residuals vector) that provides a measure of the biases affecting the measurements, to extract the redundancy information as follows:

- $\mathbf{y} = \mathbf{H} \cdot \mathbf{x} + \mathbf{e}$

- Measurements are fault-free (H_0) if $b = 0$ and at least one measurement is corrupted ($H1$) if $b \neq 0$.

- $\mathbf{y} = \mathbf{Q} \cdot \mathbf{R} \cdot \mathbf{x} + \mathbf{e}$

- $\mathbf{Q^T} \cdot \mathbf{y} = \mathbf{R} \cdot \mathbf{x} + \mathbf{Q^T} \cdot \mathbf{e}$

- $\begin{bmatrix} \mathbf{Q_1} \\ \mathbf{P} \end{bmatrix} \cdot \mathbf{y} = \begin{bmatrix} \mathbf{R_1} \\ \mathbf{0} \end{bmatrix} \cdot \mathbf{x} + \begin{bmatrix} \mathbf{Q_1} \\ \mathbf{P} \end{bmatrix} \cdot \mathbf{e}$

- Supposing that the measurement error follows a Gaussian model $\mathbf{e} \sim \mathcal{N}(\mathbf{b}, \mathbf{C})$, the parity vector is also Gaussian $\mathbf{p} \sim \mathcal{N}(\mathbf{P} \cdot \mathbf{b}, \mathbf{P} \cdot \mathbf{C} \cdot \mathbf{P^T})$

We apply RAIM to scenarios with mixed GPS and WiFi ranges considering the most general case of the Rao test to the parity vector $\mathbf{p}$ introduced in [60]. The test computes the statistic $T_R(\mathbf{p})$ and the p-value(observed significance level):

$$T_R(\mathbf{p}) = \mathbf{p}^T \cdot (\mathbf{P} \cdot \mathbf{C} \cdot \mathbf{P})^{-1} \cdot \mathbf{p} \tag{A.1}$$

where $\mathbf{C}$ is the measurement covariance matrix. We apply RAIM setting $\mathbf{C} = \mathbf{R_k}$.

In the specific case of uncorrelated measurements with the same variance: $\mathbf{C} = \sigma^2 \cdot \mathbf{I}$ and thus $T_R(\mathbf{p})$ can be simply computed in the subsequent way:

$$T_R(\mathbf{p}) = \left\|\frac{\mathbf{p}}{\sigma}\right\|^2 \tag{A.2}$$

If this p-value exceeds the predefined significance level, at least one measurement is faulty. If at least 6 anchors are available, the faulty anchor can be detected by dividing the initial set of anchors into subsets of 5 anchors and computing the p-value for each subset.

References

[1] IBeacon. http://support.apple.com/kb/HT6048,Published: Aug 4, 2017.

[2] Indoor Location Market by Component (Technology, Software Tools, and Services), Deployment Mode (Cloud, and On-premises), Application, Vertical (Transportation, Hospitality, Entertainment, Retail, and Public Buildings), and Region - Global Forecast to 2022. http://www.marketsandmarkets.com/Market-Reports/indoor-positioning-navigation-ipin-market-989.html/, year=October 2017.

[3] Microsoft indoor localization competition, http://research.microsoft.com/en-us/events/ipsn2014indoorlocalizatinocompetition/.

[4] IEEE std 802.11-2012 (revision of IEEE std 802.11-2007), IEEE standard for information technology, 2012.

[5] Ali Almagbile, Jinling Wang, and Weidong Ding. Evaluating the performances of adaptive Kalman filter methods in GPS/INS integration. *Journal of Global Positioning Systems*, 9(1):33–40, 2010.

[6] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[7] P. Bahl and V.N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In *Proc. IEEE INFOCOM'00*, Mar. 2000.

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

[9] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise Reduction in Speech Processing*, volume 2 of *Springer Topics in Signal Processing*, pages 1–4. Springer Berlin Heidelberg, 2009.

[10] Giuseppe Bianchi. Performance analysis of the ieee 802.11 distributed coordination function. *Selected Areas in Communications, IEEE Journal on*, 18(3):535–547, 2000.

[11] Jeff Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, 1998.

[12] Theodoros Bourchas, Maciej Bednarek, Domenico Giustiniano, and Vincent Lenders. Poster abstract: Practical limits of wifi time-of-flight echo techniques. In *Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, IPSN '14, pages 273–274, Piscataway, NJ, USA, 2014. IEEE Press.

[13] Roberto Calvo-Palomino, Domenico Giustiniano, Vincent Lenders, and Aymen Fakhreddine. Crowdsourcing spectrum data decoding. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE, 2017.

[14] Zhe Chen, Zhongmin Li, Xu Zhang, Guorong Zhu, Yuedong Xu, Jie Xiong, and Xin Wang. Awl: Turning spatial aliasing from foe to friend for accurate wifi localization. In *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '17, pages 238–250, New York, NY, USA, 2017. ACM.

[15] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N. Padmanabhan. Indoor localization without the pain. In *In Proc. of ACM MobiCom '10*, pages 173–184, 2010.

[16] M. Ciurana, F. Barcelo-Arroyo, and F. Izquierdo. A ranging system with IEEE 802.11 data frames. In *Radio and Wireless Symposium, 2007 IEEE*, pages 133–136, 2007.

[17] Christophe Croux and Catherine Dehon. Influence functions of the spearman and kendall correlation measures. *Statistical methods & applications*, 19(4):497–515, 2010.

[18] Davide Dardari, Andrea Conti, Ulric Ferner, Andrea Giorgetti, and Moe Z. Win. Ranging With Ultrawide Bandwidth Signals in Multipath Environments. *Proceedings of the IEEE*, 2009.

[19] Norman Richard Draper, Harry Smith, and Elizabeth Pownell. *Applied regression analysis*, volume 3. Wiley New York, 1966.

[20] J. Durbin and SJ Koopman. *Time series analysis by state space models*. Oxford University Press, Oxford, 2001.

[21] V Erceg, L Schumacher, P Kyritsi, A Molisch, DS Baum, AY Gorokhov, C Oestges, Q Li, K Yu, N Tal, et al. TGn channel models, May 2004.

[22] Aymen Fakhreddine, Domenico Giustiniano, and Vincent Lenders. Evaluation of self-positioning algorithms for time-of-flight based localization. In *2016 14th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pages 1–8. IEEE, 2016.

[23] Aymen Fakhreddine, Domenico Giustiniano, and Vincent Lenders. Data fusion for hybrid and autonomous time-of-flight positioning. In *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*, IPSN '18, pages 266–271, Piscataway, NJ, USA, 2018. IEEE Press.

[24] Aymen Fakhreddine, Nils Ole Tippenhauer, and Domenico Giustiniano. Design and large-scale evaluation of wifi proximity metrics. In *EUROPEAN WIRELESS 2018, Catania, Italy*, May 2018.

[25] D. Fernandez, F. Barcelo-Arroyo, I. Martin-Escalona, M. Ciurana, M. Jofre, and E. Gutierrez. Fusion of wlan and gnss observables for positioning in urban areas: The position ambiguity. In *Proceedings of ISCC '11*, pages 748–751. IEEE Computer Society, 2011.

[26] P. Gallo, D. Garlisi, F. Giuliano, F. Gringoli, and I. Tinnirello. WMPS: A positioning system for localizing legacy 802.11 devices. In *Transactions on Smart Processing and Computing,*, October 2012.

[27] S. Gezici, Zhi Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE Signal Process. Mag.*, 22(4):70–84, July 2005.

[28] Domenico Giustiniano, Theodoros Bourchas, Maciej Bednarek, and Vincent Lenders. Deep inspection of the noise in wifi time-of-flight echo techniques. In *Proceedings of MSWiM '15*, pages 5–12. ACM, 2015.

[29] Domenico Giustiniano and Stefan Mangold. CAESAR: Carrier sense-based ranging in off-the-shelf 802.11 wireless LAN. In *CoNEXT '11*, pages 10:1–10:12, New York, NY, USA, 2011. ACM.

[30] Jon Gjengset, Jie Xiong, Graeme McPhillips, and Kyle Jamieson. Enabling phased array signal processing on commodity wifi access points. Mobicom '14, New York, NY, USA, 2014. ACM.

[31] Stuart A. Golden and Steve S. Bateman. Sensor measurements for Wi-Fi location with emphasis on time-of-arrival ranging. *IEEE Trans. Mobile Comput.*, 6(10):1185–1198, Oct. 2007.

[32] Abhishek Goswami, Luis E. Ortiz, and Samir R. Das. Wigem: A learning-based approach for indoor localization. CoNEXT '11, pages 3:1–3:12, New York, NY, USA, 2011. ACM.

[33] Abhishek Goswami, Luis E. Ortiz, and Samir R. Das. Wigem: A learning-based approach for indoor localization. In *Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies*, CoNEXT '11, pages 3:1–3:12, New York, NY, USA, 2011. ACM.

[34] André Günther and Christian Hoene. Measuring round trip times to determine the distance between WLAN nodes. NETWORKING'05, pages 768–779, Berlin, Heidelberg, 2005. Springer-Verlag.

[35] I. Guvenc and Chia-Chin Chong. A survey on toa based wireless localization and nlos mitigation techniques. *Communications Surveys Tutorials, IEEE*, 11(3):107–124, rd 2009.

[36] Dirk Haehnel, Wolfram Burgard, Dieter Fox, Ken Fishkin, and Matthai Philipose. Mapping and localization with RFID technology. In *Proc. IEEE ICRA'05*, Apr. 2004.

[37] Ulrich Hammes and Abdelhak M Zoubir. Robust mt tracking based on m-estimation and interacting multiple model algorithm. *Signal Processing, IEEE Transactions on*, 59(7):3398–3409, 2011.

[38] Suining He and S-H Gary Chan. Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons. *IEEE Communications Surveys & Tutorials*, 18(1):466–490.

[39] Kiran Joshi, Steven Hong, and Sachin Katti. Pinpoint: Localizing interfering radios. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, nsdi'13, pages 241–254, Berkeley, CA, USA, 2013. USENIX Association.

[40] Elliott Kaplan and Christopher Hegarty. *Understanding GPS: principles and applications*. Artech house, 2005.

[41] John Klobuchar et al. Ionospheric time-delay algorithm for single-frequency gps users. *Aerospace and Electronic Systems, IEEE Transactions on*, (3):325–331, 1987.

[42] Sadanori Konishi and Genshiro Kitagawa. *Information criteria and statistical modeling*. Springer Science & Business Media, 2008.

[43] Aleksandra Korolova and Vinod Sharma. Cross-app tracking via nearby bluetooth devices. In *Proceedings of PrivacyCon*, 2017.

[44] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, pages 269–282, New York, NY, USA, 2015. ACM.

[45] Steven Lanzisera, David Zats, and Kristofer S. J. Pister. Radio Frequency Time-of-Flight Distance Measurement for Low-Cost Wireless Sensor Localization. *IEEE Sensors Journal*, 2011.

[46] Joon-Yong Lee and Robert A. Scholtz. Ranging in a Dense Multipath Environment Using an UWB Radio Link. *IEEE Journal on Selected Areas in Communications*, 2002.

[47] Michael Levandowsky and David Winter. Distance between sets. *Nature*, 234(5323):34–35, 1971.

[48] Xinrong Li and K. Pahlavan. Super-resolution toa estimation with diversity for indoor geolocation. *Wireless Communications, IEEE Transactions on*, 3(1):224–234, Jan 2004.

[49] Xinrong Li, K. Pahlavan, M. Latva-aho, and M. Ylianttila. Comparison of indoor geolocation methods in dsss and ofdm wireless lan systems. In *IEEE Fall VTC 2000*, volume 6, pages 3015–3020 vol.6, 2000.

[50] Hyuk Lim, Lu-Chuan Kung, J.C. Hou, and Haiyun Luo. Zero-configuration, robust indoor localization: Theory and experimentation. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, April 2006.

[51] Hui Liu, H. Darabi, P. Banerjee, and Jing Liu. Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1067–1080, Nov 2007.

[52] Dimitrios Lymberopoulos, Jie Liu, Xue Yang, Romit Roy Choudhury, Vlado Handziski, and Souvik Sen. A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, IPSN '15, pages 178–189, New York, NY, USA, 2015. ACM.

[53] Alex T. Mariakakis, Souvik Sen, Jeongkeun Lee, and Kyu-Han Kim. Sail: Single access point-based indoor localization. In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '14, pages 315–328, New York, NY, USA, 2014. ACM.

[54] D.D. McCrady, L. Doyle, H. Forstrom, T. Dempsey, and M. Martorana. Mobile ranging using low-accuracy clocks. *Microwave Theory and Techniques, IEEE Transactions on*, 48(6):951–958, 2000.

[55] Pratap Misra and Per Enge. *Global Positioning System: Signals, Measurements and Performance Second Edition*. Lincoln, MA: Ganga-Jamuna Press, 2006.

[56] Rajalakshmi Nandakumar, Swati Rallapalli, Krishna Chintalapudi, Venkata N Padmanabhan, Lili Qiu, Aishwarya Ganesan, Saikat Guha, Deepanker Aggarwal, and Aakash Goenka. Physical analytics: A new frontier for (indoor) location research. 2013.

[57] Clement A Ogaja. *Geomatics Engineering: A Practical Guide to Project Design*. CRC Press, 2016.

[58] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Proceedings of MobiSys '10*, pages 299–314. ACM, 2010.

[59] Bradford W Parkinson, James J Spilker Jr, et al. Global positioning system: Theory and applications volume 1 & ii, aiaa, 1996.

[60] Lucila Patino-Studencka, Günter Rohmer, and Jörn Thielecke. Approach for detection and identification of multiple faults in satellite navigation. In *PLANS, 2010 IEEE/ION*, pages 221–226. IEEE, 2010.

[61] Eldad Perahia and Robert Stacey. *Next generation wireless LANs: 802.11 n and 802.11 ac*. Cambridge university press, 2013.

[62] Paolo Pettinato, Niklas Wirstrom, Joakim Eriksson, and Thiemo Voigt. Multi-Channel Two-way Time of Flight Sensor Network Ranging. EWSN 2012.

[63] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proc. ACM MobiCom'00*, Aug. 2000.

[64] Anshul Rai, Krishna Kant Chintalapudi, Venkata N. Padmanabhan, and Rijurekha Sen. Zee: Zero-effort crowdsourcing for indoor localization. Mobicom '12, pages 293–304, New York, NY, USA, 2012. ACM.

[65] Maurizio Rea, Aymen Fakhreddine, Domenico Giustiniano, and Vincent Lenders. Filtering noisy 802.11 time-of-flight ranging measurements from commoditized wifi radios. *IEEE/ACM Transactions on Networking*, 25(4):2514–2527, 2017.

[66] Souvik Sen, Jeongkeun Lee, Kyu-Han Kim, and Paul Congdon. Avoiding multipath to revive inbuilding wifi localization. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '13, pages 249–262, New York, NY, USA, 2013. ACM.

[67] Skyhook. `http://www.skyhookwireless.com/`.

[68] Wei Sun, Junliang Liu, Chenshu Wu, Zheng Yang, Xinglin Zhang, and Yunhao Liu. Moloc: On distinguishing fingerprint twins. In *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*, pages 226–235. IEEE, 2013.

[69] F. Tappero and et al. IEEE 802.11 ranging and multi-lateration for software-defined positioning receiver. In *IPIN'10*. IEEE, 2010.

[70] I. Tinnirello, Sunghyun Choi, and Youngsoo Kim. Revisit of RTS/CTS exchange in high-speed IEEE 802.11 networks. In *WoWMoM*, pages 240–248, June 2005.

[71] Ilenia Tinnirello, Domenico Giustiniano, Luca Scalia, and Giuseppe Bianchi. On the side-effects of proprietary solutions for fading and interference mitigation in IEEE 802.11b/g outdoor links. *Comput. Netw.*, 53(2):141–152, Feb. 2009.

[72] Qinghua Wang, Ilangko Balasingham, Miaomiao Zhang, and Xin Huang. Improving rss-based ranging in los-nlos scenario using gmms. *Communications Letters, IEEE*, 15(10):1065–1067, 2011.

[73] Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. The active badge location system. *ACM Trans. Inf. Syst.*, 10(1):91–102, Jan. 1992.

[74] Hongkai Wen, Zhuoling Xiao, Niki Trigoni, and Phil Blunsom. On assessing the accuracy of positioning systems in indoor environments. In *Proceedings of EWSN'13*, pages 1–17. Springer-Verlag, 2013.

[75] Anthony J Wheeler, Ahmad Reza Ganji, Vaidyanadhan Venkata Krishnan, and Brian S Thurow. *Introduction to engineering experimentation*. Prentice Hall New Jersey, 1996.

[76] Erik Wilhelm, Sandra Siby, Yuren Zhou, Xavier J. S. Ashok, Melani Jayasuriya, Shaohui Foong, Jacksheeng Kee, Kristin Wood, and Nils Ole Tippenhauer. Wearable environmental sensors and infrastructure for mobile large-scale urban deployment. *Sensors*, 16(22):8111–8123, November 2016.

[77] Zhuoling Xiao, Hongkai Wen, Andrew Markham, Niki Trigoni, Phil Blunsom, , and Jeff Frolik. Identification and mitigation of non-line-of-sight conditions using received signal strength. In *Proceedings of the 9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2013)*, Lyon, France, October 2013.

[78] Jie Xiong and Kyle Jamieson. Arraytrack: A fine-grained indoor location system. In *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, pages 71–84, Lombard, IL, 2013. USENIX.

[79] Jie Xiong, Karthikeyan Sundaresan, and Kyle Jamieson. Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization. In *MobiCom '15*, pages 537–549, New York, NY, USA, 2015. ACM.

[80] Moustafa Youssef and Ashok Agrawala. The horus wlan location determination system. MobiSys '05, pages 205–218, New York, NY, USA, 2005. ACM.

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