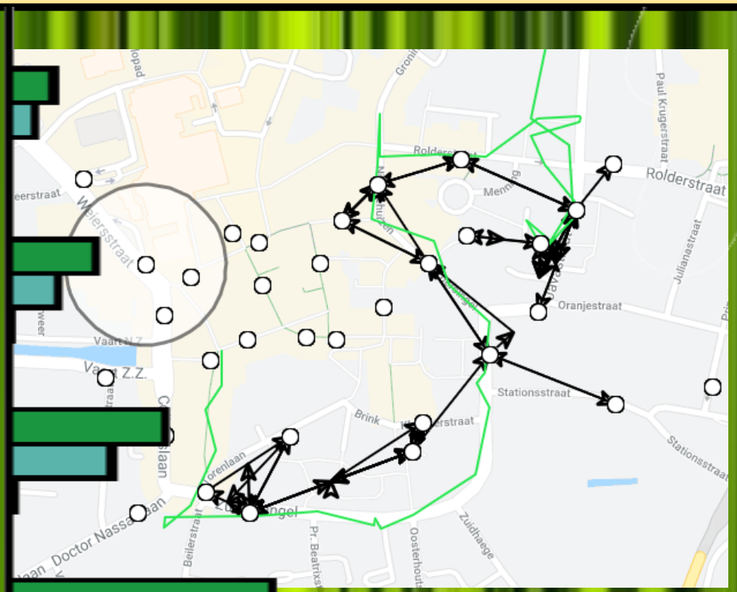
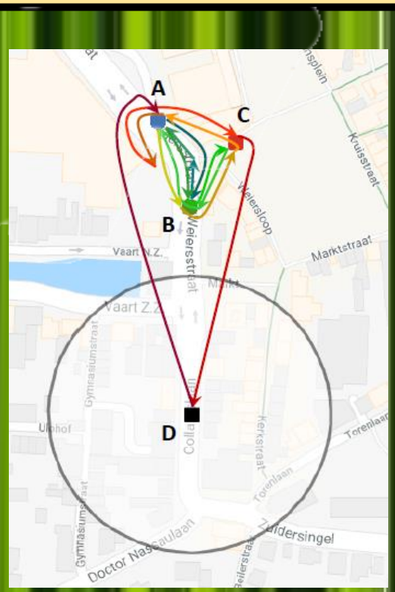
CROWD DATA ANALYTICS AS SEEN FROM WIFI

A CRITICAL REVIEW





Monitoring and modelling crowd movement enables a plethora of applications. Crowd-movement analysis has classically been done manually, only at large scales (spatial and temporal) and based on small samples. By automating the process, we can dramatically increase the sample size, the amount of data. WiFi remote-positioning is currently the most popular technology to achieve this goal. However, not enough research has been conducted in order to understand the quality of the data generated through WiFi remote-positioning. This thesis aims to address the issue and raise a warning light regarding the technology.

CROWD DATA ANALYTICS AS SEEN FROM WIFI Cristian Chilipirea

ISBN: 978-90-365-4896-0
DOI: 10.3990/1.9789036548960

Cristian Chilipirea

CROWD DATA ANALYTICS AS SEEN FROM WIFI

A CRITICAL REVIEW

Cristian Chilipirea

This dissertation has been approved by:
Supervisors:

Prof. Dr. Ir. M.R. van Steen	University of Twente, The Netherlands
Prof. Dr. V. Cristea	University Politehnica of Bucharest, Romania
Prof. Dr. C. Dobre	University Politehnica of Bucharest, Romania
Dr. M. Baratchi	Leiden University, The Netherlands

Cover design:	Cristian Chilipirea
ISBN:	978-90-365-4896-0
DOI:	10.3990/1.9789036548960

CROWD DATA ANALYTICS AS SEEN FROM WIFI
A CRITICAL REVIEW

DISSERTATION

to obtain a joint degree, namely
the degree of doctor at the Universiteit Twente
on the authority of the rector magnificus of the University of Twente,
Prof. Dr. T.T.M. Palstra,
and at the University Politehnica of Bucharest,
on the authority of the rector of University Politehnica of Bucharest,
Prof. Dr. M.C. Costoiu,
on account of the decision of the graduation committee
to be publicly defended
on Thursday 21 November 2019 at 16:45

by

Cristian Chilipirea
born on 7th of June 1989
in Bucharest, Romania

Graduation Committee:

Chairman / secretary	Prof. Dr. P.J.F. Lucas
Supervisor:	Prof. Dr. M.R. van Steen
	Prof. Dr. V. Cristea
	Prof. Dr. C. Dobre
	Dr. M. Baratchi
Committee Members:	Prof. Dr. P. Shenoy
	Prof. Dr. S. Klous
	Prof. Dr. J.L. van den Berg
	Dr. N. Meratnia
	Dr. A. Peter
	Prof. Dr. Ir. M.R. van Steen
	Dr. M. Baratchi
	Prof. Dr. C. Dobre
	Prof. Dr. V. Cristea

Acknowledgments

First of all, I would like to thank the defense committee for the thorough and invaluable feedback which they provided. Prof. Prashant Shenoy, Prof. Sander Klous, Prof. Hans van den Berg, Dr. Nirvana Meratnia, Dr. Andreas Peter took time out of their busy schedules to read and review my work, as well as attend the defense ceremony. For this I cannot be grateful enough. To this list I would like to add Prof. Peter Lucas, who stands as president of the defense committee.

It has been six years since I first met Prof. Maarten van Steen. He guided me through my master's thesis and continued to do so through my PhD. I've read many stories about meeting that one amazing person and against all odds and hopes convincing said person to become their mentor. For me that person is Prof. Maarten van Steen. To say that Prof. Maarten van Steen molded me into a researcher would be an understatement. It took five years but now I know not to leave a path untraveled or stone unturned.

Some say it is difficult to work for two bosses, but I ended up with three and somehow everything was better for it. If Prof. Maarten van Steen taught me what it means to be serious, punctual and precise, Prof. Valentin Cristea and Prof. Ciprian Dobre engraved in me what is any researcher's creed of "publish or perish". Prof. Ciprian Dobre is also the reason why I returned to Romania after I finished my master's, mostly because of his energy and unrelenting attitude. Furthermore, Prof. Valentin Cristea offered the model of what an academic should be like for his students. His class of Parallel and Distributed Algorithms has become my obsession.

With such incredible mentors it is difficult to make mistakes. But if anything slipped through the cracks it was caught by Dr. Mitra Baratchi. She always offered advice and reviewed every idea and every text I wrote.

Whenever I was stressed, wanted to give up, or generally had an issue, there was one person who I knew I could count on for advice. Whatever the hour Prof. Florin Pop would answer his phone call and would always provide the most appropriate answer and guide me towards the most diplomatic approach.

This entire work is based on data and a lot of data. Most of it would not

have been available if it wasn't for Roel Schiphorst from BlueMark Innovations and Jeroen van Ingen from University of Twente. I think I still owe a beer to Jeroen.

PhD candidate Valeriu Stanciu is following this same path and many times we had the same issues and same struggles. To him I wish all the luck.

During these years I have spent time discussing with a large variety of people. Some of these discussions generated ideas and some papers. For anyone whom I didn't explicitly mention, I would like to express my gratitude for the time they shared with me.

Looking back, I started this trip without really knowing what I was getting into. I was young, a bachelor and everything was possible. In the meantime, I became a husband to Andreea and more recently a father to Laura. Getting a PhD is such a difficult task that only crazy people try their hand at it. Pursuing a PhD while raising a baby is ludicrous and should generally be considered unsafe and unhealthy.

For me, it is difficult to feel pride for a finished project. Laura guarantees that I will feel nothing but pride when looking back. By the way, she reminded me how simple it is to smile for no reason and be happy all day long. No one can resist a child laugh or smile, and neither can I.

None of this would have been possible without my wife Andreea. She started on the same path and sacrificed her time to make sure I would get through it. She put up with me in more ways than one can imagine. I will never be able to repay her for all she did and all she gave me. Love is not a strong enough word to describe what bounds me to her. It is beyond a pledge or an oath that I will always be here for her.

Cristian Chilipirea
Bucharest, October 2019

Abstract

Monitoring and modeling crowd movement enables a plethora of applications. Understanding crowd dynamics can help us enhance our cities by enabling improved facility planning and by directing better policies. Crowd monitoring can help prevent disasters and for those that happened, assist and improve with response. Furthermore, many commercial applications spanning from business analytics to marketing, to name just a few examples, make use of crowd monitoring while many more are being added as we develop smart cities.

Crowd-movement analysis has classically been done manually, only at large scales (spatial and temporal) and based on small samples. By automating the process, we can dramatically increase the sample size, the amount of data, and as such be able to infer granular movements, previously unmeasurable. This allows us to build better crowd-dynamics models. Many technologies have appeared that automate mobile-data gathering. Out of these, WiFi remote-positioning (a technique for using a set of sensors to record positions of all individuals carrying WiFi devices, such as smartphones) appears to be the most recent and popular as it promises to offer a balance between deployment price, the crowd's size (number of individuals) of what can be monitored, and positional accuracy.

We have studied the existing literature and conducted our own WiFi remote-positioning data-gathering experiments in order to understand the completeness, or lack thereof, and granularity of movements that can be described using the technology. We focus on understanding what are the benefits and which are the limitations of WiFi remote-positioning by decreasing the size of the covered area to a city center (or campus), and the time period to that of a day. This restricts us to observing short movements, such as going to work, shopping or moving between classes, as a few examples. Our self-imposed restrictions follow from our concern for preserving privacy. This can be translated into our main research question: **To what extent can we model crowd dynamics based on current positioning technologies?**

Positioning based on WiFi is known to be biased as it cannot be used for individuals that do not carry WiFi devices. On top of that, our analysis shows that the information extracted from WiFi remote-positioning data sets is underwhelming compared to the public attention that surrounds the technology This is based on several different data sets that we collected. Detections are sparse and low spatial accuracy introduces difficult to circumvent anomalies that hide detailed movements. For most detected devices, we do not have enough data to identify even a single movement. For others, we can trace only few movements. Most movements are hidden by anomalies that resemble a movement in circles.

In order to mitigate the anomalies, we have developed and extensively measured the effectiveness of techniques to smooth traces as well as methods to extract information from positioning data in the form of stops and moves. Although these techniques managed to improve the quality of the data and make it more usable, there are limits to how effective they were.

Our attempts to improve the results by adding more sensors backfired. Not only did the amount of information not increase by adding more sensors, but we also discovered we could obtain the same results with fewer. This has the advantage of potentially lowering the financial cost for deploying WiFi remote-positioning platforms.

We explored the use of alternative data sources for WiFi remote positioning, as opposed to the widely adopted use of Probe Request frames (a specific data packet transmitted by WiFi devices). Analysis of positions based on WiFi connection logs showed that they contain a significant amount of information not extracted by most WiFi remote-positioning platforms. This raises questions about the bias of WiFi remote-positioning deployments. As our research uncovered, it is likely that many WiFi remote-positioning data sets do not include positioning data for periods when devices are connected to a network.

Samenvatting

Monitoring en modellering van menigtebewegingen maakt een overvloed aan toepassingen mogelijk. Inzicht in de dynamiek van de menigte kan ons helpen onze steden te verbeteren door een betere planning van de faciliteiten mogelijk te maken en door een beter beleid te sturen. Monitoring van mensenmassa's kan rampen helpen voorkomen en voor degenen die zijn gebeurd, helpen en verbeteren met respons. Bovendien maken veel commerciële toepassingen, variërend van bedrijfsanalyse tot marketing, om maar een paar voorbeelden te noemen, gebruik van monitoring van voetgangers terwijl er nog veel meer worden toegevoegd bij het ontwikkelen van slimme steden.

Bewegingsanalyse van voetgangers is doorgaans altijd handmatig gedaan, alleen op grote schaal (ruimtelijk en tijdelijk) en op basis van kleine steekproeven. Door het proces te automatiseren, kunnen we de steekproefomvang en de hoeveelheid gegevens drastisch vergroten en zo granulaire bewegingen afleiden die eerder onmeetbaar waren. Hiermee kunnen we betere modellen bouwen. Er zijn veel technologieën verschenen die het verzamelen van mobiele gegevens automatiseren. Hiervan lijkt WiFi-positionering op afstand (een techniek voor het gebruik van een verzameling sensoren voor het opnemen van posities van alle personen met WiFi-apparaten, zoals smartphones) de meest recente en populaire omdat het belooft een evenwicht te bieden tussen de prijs, de grootte van de menigte (aantal personen) van wat kan worden gemonitord, en positionele nauwkeurigheid.

We hebben de bestaande literatuur bestudeerd en onze eigen experimenten voor het verzamelen van gegevens op afstand op basis van WiFi uitgevoerd om de volledigheid of het gebrek daaraan en de granulariteit van bewegingen te begrijpen die met behulp van de technologie kunnen worden beschreven. We richten ons op het begrijpen van de voordelen en de beperkingen van WiFi-positionering op afstand door de grootte van het overdekte gebied tot een stadscentrum (of campus) te verkleinen, en de tijdsperiode tot die van een dag. Dit beperkt ons tot het observeren van korte bewegingen, zoals naar het werk gaan, winkelen of tussen colleges gaan, als een paar voorbeelden. Onze zelfopgelegde

beperkingen vloeien voort uit onze zorg voor het behoud van privacy. Dit kan worden vertaald in onze hoofdvraag: textbf In hoeverre kunnen we mensen massa's modelleren op basis van huidige positioneringstechnologieën?

Het is bekend dat positionering op basis van WiFi bevooroordeeld is, omdat deze niet kan worden gebruikt voor personen die geen WiFi-apparaten hebben. Bovendien blijkt uit onze analyse dat de informatie die is geëxtraheerd uit de gegevens voor positionering op afstand via WiFi, overweldigend is in vergelijking met de publieke aandacht voor de technologie. Dit is gebaseerd op verschillende gegevens die we hebben verzameld. Detecties zijn schaars en lage ruimtelijke nauwkeurigheid introduceert moeilijk te omzeilen afwijkingen die gedetailleerde bewegingen verbergen. Voor de meeste gedetecteerde apparaten hebben we onvoldoende gegevens om zelfs maar één beweging te identificeren. Voor anderen kunnen we slechts enkele bewegingen traceren. De meeste bewegingen worden verborgen door anomalieën die lijken op een beweging in cirkels.

Om de afwijkingen te verminderen, hebben we de effectiviteit van technieken om sporen te verzachten en methoden om informatie uit positiegegevens te extraheren in de vorm van stops en bewegingen, ontwikkeld en uitgebreid gemeten. Hoewel deze technieken erin geslaagd zijn om de kwaliteit van de gegevens te verbeteren en bruikbaarder te maken, zijn er grenzen aan hoe effectief ze waren.

Onze pogingen om de resultaten te verbeteren door meer sensoren achteraf toe te voegen. Niet alleen nam de hoeveelheid informatie niet toe door meer sensoren toe te voegen, maar we ontdekten ook dat we met minder dezelfde resultaten konden bereiken. Dit heeft het voordeel dat de financiële kosten voor het gebruik van externe WiFi-positioneringsplatforms mogelijk worden verlaagd.

We hebben het gebruik van alternatieve gegevensbronnen voor WiFi-positionering op afstand onderzocht, in tegenstelling tot het alom geaccepteerde gebruik van Probe Request-frames (een specifiek datapakket verzonden door WiFi-apparaten). Analyse van posities op basis van WiFi-verbindingslogboeken toonde aan dat deze een aanzienlijke hoeveelheid informatie bevatten die niet werd geëxtraheerd door de meeste externe WiFi-positioneringsplatforms. Dit roept vragen op over de vertekening van implementaties van WiFi-positionering op afstand. Zoals ons onderzoek aan het licht heeft gebracht, is het waarschijnlijk dat veel WiFi-gegevens voor positionering op afstand geen plaatsbepalingsgegevens bevatten gedurende perioden waarin apparaten zijn verbonden met een netwerk.

Abstract

Monitorizarea și modelarea mișcării mulțimilor permite o multitudine de aplicații. Înțelegerea dinamicii mulțimilor ne poate ajuta să îmbunătățim orașele, permițând o eficientizare a planificării infrastructurii și direcționând politici mai bune. Monitorizarea mulțimilor poate ajuta la prevenirea gestionarea dezastrelor prin îmbunătățirea timpului de răspuns. Mai mult, numeroase aplicații comerciale, de la analiza business-ului până la marketing, pentru a numi doar câteva exemple, se folosesc monitorizarea mulțimilor. Alte aplicații se dezvoltă pe măsură ce dezvoltăm orașe inteligente – smart cities.

Analiza mișcării mulțimilor a fost executată manual, la scară mare (atât spațial cât și temporal) și pe baza unor seturi mici de date. Prin automatizarea procesului, putem crește dramatic dimensiunea eșantionului, cantitatea de date și, astfel, putem deduce mișcări granulare, care anterior nu au putut fi măsurate. Acest lucru ne permite să construim modele mai bune de dinamică a mulțimilor. Recent, au apărut multe tehnologii care automatizează colectarea datelor mobile. Dintre acestea, poziționarea la distanță efectuată prin WiFi (o tehnică pentru utilizarea unui set de senzori pentru a înregistra pozițiile tuturor persoanelor care transportă dispozitive WiFi, cum ar fi telefoanele inteligente - smartphone-urile) pare a fi cea mai populară, deoarece promite să ofere un echilibru între costul unei astfel de platforme, dimensiunea mulțimii (numărul de indivizi) a ceea ce poate fi monitorizat și precizia pozițională.

Am studiat literatura existentă și am realizat propriile noastre experimente de colectare a datelor de la distanță folosind WiFi pentru a înțelege completitudinea datelor, sau lipsa acestora, și granularitatea mișcărilor care pot fi descrise folosind această tehnologie. Ne concentrăm pe a înțelege care sunt avantajele și care sunt limitările poziționării la distanță folosind WiFi prin scăderea dimensiunii zonei monitorizate la cea a unui centru de oraș (sau campus) și perioada de timp până la cea a unei zile. Acest lucru ne restricționează să observăm mișcări scurte, cum ar fi mersul la muncă, cumpărăturile sau mutarea între clase, ca fiind exemple. Restricțiile noastre autoimpuse rezultă din preocuparea noastră pentru garantarea vieții private a persoanelor care sunt monitorizate. Acest

lucru poate fi tradus în principala noastră întrebare de cercetare: **În ce măsură putem modela dinamica mulțimilor bazată pe tehnologiile de poziționare actuale?**

Poziționarea bazată pe WiFi este cunoscută ca oferind date incomplete, deoarece nu poate fi utilizată pentru monitorizarea persoanele care nu poartă dispozitive WiFi. În plus, analiza noastră arată că informațiile extrase din seturile de date de poziționare de la distanță folosind WiFi sunt dezamăgitoare în comparație cu atenția publică care înconjoară tehnologia. Această concluzie se bazează pe analiza mai multor seturi de date, foarte diferite, pe care le-am colectat. Detecțiile rare și precizia spațială scăzută introduce anomalii dificil de evitat care ascund mișcări detaliate. Pentru majoritatea dispozitivelor detectate, nu avem suficiente date pentru a identifica nici măcar o singură mișcare. Pentru altele, putem urmări doar puține mișcări. Majoritatea mișcărilor sunt ascunse de anomalii care seamănă cu o plimbare în cercuri.

Pentru a atenua anomaliile, am dezvoltat și am măsurat pe larg eficacitatea tehnicilor pentru simplificarea traseelor, precum și metode pentru extragerea informațiilor sub formă de opriri și mișcări. Deși aceste tehnici au reușit să îmbunătățească calitatea datelor și să le facă mai utilizabile, există limite asupra cât de eficiente au fost acestea.

Încercările noastre de a îmbunătăți rezultatele adăugând mai mulți senzori au eșuat. Nu numai că nu am crescut cantitatea de informație prin adăugarea mai multor senzori, dar am descoperit, de asemenea, că putem obține aceleași rezultate cu mai puține. Această descoperire prezintă avantajul de a reduce potențial costurile financiare pentru implementarea platformelor de poziționare la distanță folosind WiFi.

Am explorat utilizarea surselor de date alternative pentru poziționarea la distanță folosind WiFi, spre deosebire de utilizarea pe scară largă a cadrelor de tip Probe Request (un pachet de date specific transmis de dispozitivele WiFi). Analiza pozițiilor bazate pe jurnalele de conexiune WiFi a arătat că acestea conțin o cantitate semnificativă de informații care nu sunt extrase de majoritatea platformelor de poziționare la distanță folosind WiFi. Acest lucru ridică întrebări cu privire la rezultatele implementărilor de poziționare la distanță folosind WiFi. În concluzie, este foarte probabil ca multe seturi de date de poziționare la distanță folosind WiFi să nu includă date de poziționare pentru perioadele în care dispozitivele sunt conectate la o rețea.

Contents

CHAPTER 1

Introduction

Mobility has influence on a large variety of factors that affect human life [1]. A prime example would be the shape, size, and feel of our cities. These features are dictated by the dynamics of inhabitants. Cities have evolved throughout history, in an organic way, remaining in par with transportation technologies. Considering this, it comes as no surprise that urban and facility planning is heavily concerned with mobility.

It is not only the architecture of our cities that is affected by mobility, but also geopolitics and, in turn, our economic and social structures. Furthermore, human mobility has a direct impact on the environment, for example through pollution produced by cars or planes. Even our safety and security is swayed by mobility through events (such as crowds trying to get out of a burning building) or biologic factors (such as the spread of diseases through a population).

The advent of increasing feasibility of automatically gathering and analyzing urban data has led to what are generally called smart cities. Data on pedestrian dynamics is an important component of urban data. Concentrating on mobility, we can imagine living in cities where the transportation becomes more efficient and adapts to the real-time needs of the inhabitants; where the schedule of businesses or public institutions changes in order to make them available so that they can serve the largest number of people; where during emergencies the flows of people are optimized so that the biggest number of lives are saved; where search and rescue has tools that permit them to best utilize their resources; where we build stronger, more inclusive communities; where energy is saved and pollution is reduced through fine control of our utilities (e.g. street illumination).

Facility planning, smart cities, marketing, tourism and entertainment are just a few examples of fields that can benefit from understanding mobility, or more precisely, the dynamics of crowds. As such, monitoring and modeling crowd dynamics becomes more important than ever. All the applications we described previously are dependent, or can be improved, given crowd-dynamics infor-

mation. Information which so far has been gathered using slow and inefficient means, such as having someone count or manually track people.

Crowd dynamics can be represented by the total of position changes for all, or a sample of individuals. This type of data is relevant only at the level of crowds or groups of people. However, classical positioning technologies, such as GPS, are aimed at the individual. They are intrusive and raise important privacy concerns. What is worse is that this intrusiveness makes them impossible to scale to large crowds.

The popularity of smartphones and the wide adoption of a handful of communication protocols potentially enables nonintrusive positioning for large masses of individuals. These technologies are intrinsically privacy sensitive when used for positioning compared to other methods, such as the use of video recordings (we will discuss this more in the next chapter).

Although multiple companies, applications and significant research makes use of these positioning and monitoring technologies based on existing communication protocols, their outputs are not completely understood. This brings us to our main research question:

To what extent can we model outdoor crowd dynamics based on current positioning technologies?

1.1 Contributions

Our main research question can be broken into several smaller ones. Firstly, to determine the extent to which we can model crowd dynamics we need to identify the most suited positioning technology. This brings us to the first research question:

Question 1: *Which positioning technology can be used to provide the highest amount of data for the highest number of individuals, and, as such, is best suited for monitoring crowd dynamics?*

• **Chapter 2:** To answer our first question ***we conduct a survey of positioning systems***. Positioning data has a large variety of applications and no available solution is perfect or suitable for all. For example, GPS, the most popular positioning system, does not work indoors. This has triggered the implementation of multiple alternatives, each with advantages and disadvantages.

Our survey shows WiFi remote positioning as the most promising technology for crowd-dynamics analysis due to the relative ease at which it automatically collects data in a nonintrusive manner. This allows it to scale to large amounts of data for many individuals. Having this answer, we can address four ques-

tions (2, 3, 5 and 6), which combined offer a response our main research problem.

Question 2: *How is WiFi remote positioning implemented and what are the current applications it is used for?*

• **Chapter 2:** To gain an insight in the capabilities of this technology ***we conduct a survey on current applications of WiFi remote positioning*** and ***implement our own WiFi remote-positioning systems***. During the implementation we discover essential details, in the form of possible configuration parameters and properties of the resulting data, that have not been thoroughly explored in the literature.

Our description of WiFi remote-positioning methods are based on our experiences with WiFi crowd-dynamics monitoring platforms. ***We conducted five data-gathering experiments in three cities*** resulting in data sets that describe cumulatively the movements of hundreds of thousands of individuals for a time period of a month. The answer to the next research question is based on these data sets and our experiences.

Question 3: *What are the properties of traces extracted from data produced by WiFi remote-positioning systems?*

• **Chapter 2:** ***We conduct analysis on the data set, both at an aggregated and at a per-trace level***. During this analysis and based on visualization of traces we observed that WiFi remote positioning generates traces that are sparse and contain various anomalies. This brings another question (4).

Question 4: *Why are the traces sparse and what are the cyclic-movement anomalies we observe? How can we mitigate the effect caused by said anomalies?*

• **Chapter 3:** In order to understand the sparsity and anomalies we start by ***analyzing basic properties of the WiFi remote-positioning technology***. We go into details on the positional accuracy and frequency of detections. ***We show that these properties cause traces to contain an abundance of anomalies that can best be described as "moving in circles".*** These anomalies are not particular to WiFi remote positioning but are also common for traces obtained with different technologies, such as GPS. However, the anomalies are more problematic for WiFi remote positioning as they appear at a much larger scale. ***We develop three solutions that smooth traces, which can be used to manage the anomalies***. Alongside, ***we develop metrics based on entropy and dissimilarity that describe the effectiveness of our smoothing algorithms***.

Question 5: *What useful information can be extracted from positioning data in order*

to build crowd-dynamics models and how can we quantify this information?

• **Chapter 4:** Crowd-dynamics models require movement information from many individual traces. However, a trace may contain many superfluous data points. Because of this, the amount of information on crowd dynamics cannot be correlated to the amount of data generated by the positioning technology. An extensive research through the existing literature has revealed periods of stops of moves to be the most relevant type of information for crowd-dynamics models.

We identified and adapted algorithms developed to extract information from GPS traces (algorithms that identify periods of stops and moves) to work with WiFi remote-positioning data sets. The resulting sets of stops and moves can be used to describe crowd dynamics in a simple and concise way. This enables their use in conducting complex analyses. Furthermore, stops and moves represent the total information that can be extracted from WiFi remote positioning traces. Using the number of stops and moves as a metric, we can address our last questions.

Question 6: *How much crowd-dynamics information can we extract using WiFi remote-positioning and how can we increase this value?*

In order to increase the amount of crowd-dynamics information, we explore two possibilities: the effect of the number of sensors and the implementation of an alternative data source based on WiFi. These are addressed in the final questions (7 and 8).

Question 7 (part of question 6): *Can we increase the amount of crowd-dynamics information by adding more sensors and as such, increasing the amount of positioning data?*

• **Chapter 5:** This question is particularly important because a linear correlation between the amount of positioning data and the amount of information means that any platform based on this technology can be improved given a higher cost, by simply adding more sensors. ***We study the effect that the density of sensors has on the set of stops and moves that describe crowd dynamics***. As stated previously, the set of stops and moves is representative of the amount of information that can be extracted from WiFi crowd-dynamics data.

Question 8 (part of question 6): *Can we increase the amount of crowd-dynamics information by using alternative WiFi data sources?*

• **Chapter 6:** Most WiFi remote-positioning data sets are gathered by recording Probe Request frames (described in Chapter 2). This was also our initial

approach, after studying the literature. Later, we discovered that positioning data can also be successfully obtained from WiFi connection logs.

We conducted a data gathering experiment where we recorded both Probe Requests and connection logs. In order to fully understand the extent to which we can model crowd dynamics based on WiFi remote-positioning data we need to address the problem of completeness. If these two data sets do not offer the same information, it means each individually is not complete. We know that we cannot monitor people who do not carry a communication device (WiFi in our case), but it is not clear how complete is the information extracted for the other cases.

We compare the two WiFi remote-positioning data sets. Based on the differences we show that in most cases more positioning data could have been gathered and the amount of information increased. This raises questions about how representative data gathered with WiFi remote positioning is for modeling crowd dynamics.

1.2 Technical Overview

Positioning is the process of discovering a **target's** location relative to one, or multiple, **reference points** (also called **anchors**). By recording timestamped positions, we can then **trace** the movement of a target.

In the case of the Global Positioning System [2] (GPS), the most popular positioning method and the first implementation of a Global Navigation Satellite System (GNSS), satellites[1] are used as **reference points**. The position of the **target** is calculated relative to the satellites and converted to one in the geographic coordinate system [3] (latitude, longitude, and altitude). The conversion is done by combining the target's relative position to the satellites with the position of the satellites on the geographic coordinate system. The position of the satellites is known, although continuously changing[2]. The position changes because the satellites are not geostationary, meaning their orbits do not match the rotation speed of the Earth.

A target's position can be determined by the target itself (**self-positioning**), or it can be determined by external entities, possibly the anchors (**remote positioning**). If the target is not involved in the positioning process, determining its location can be difficult. The target can actively help or undermine other entities from finding it.

[1] `https://www.gps.gov/systems/gps/space/` (accessed April 3, 2019)
[2] `https://www.n2yo.com/satellites/?c=20` (accessed April 3, 2019)

Positioning is critical to us as individuals and to the multiple systems that we built which depend on a form of it. However, positioning has always been difficult (consider finding your way through a new city without GPS, or maps). Because of its importance, difficulty and the fact that no solution can serve all requirements, a lot of different positioning technologies have been developed. Positioning can be achieved by systems that use visual [4], magnetic [5], inertial [6], electromagnetic [7], acoustic [8] or even olfactory [9] data.

Radio-signal positioning systems work by having the anchors or the target transmit electromagnetic signals, which are received and used by the other party. The signals can carry information that can help improve the accuracy of positioning, like in the case of GPS.

In recent years a new class of radio-signal positioning systems has appeared. These systems are based on existing and well-established communication protocols. There are Bluetooth positioning systems [10], WiFi positioning systems [11], GSM positioning systems [12] and 4G positioning systems [13]. These systems make use of signals that are already widely used. Smartphones have all these communication capabilities and are with us all the time. By discovering the position of a smartphone (or similar mobile devices) we discover the position of the individual carrying it.

Positioning based on communication protocols can take the form of self and remote positioning. Self-positioning is done by the mobile device (**target**) which receives signals from access points (**anchor**). Remote positioning is done by an external system or device recording the signals generated by the mobile device.

Because of the prevalence of smartphones, communication protocols can be used to do remote positioning on large numbers of individuals. This is possible because these positioning systems make use of the signals already transmitted by smartphones. This means that the target devices do not have to be involved, they can be passive and require no modification to their software or hardware. Alternative techniques to gather positioning data from individuals (traditionally based on GPS) require their involvement and because of this they become intrusive and do not scale to many people.

WiFi positioning is a form of radio-signal positioning that uses signals standardized in the WiFi 802.11 communication protocol family [14]. WiFi signals are organized as frames and are transmitted and received by both mobile devices, such as smartphones, tablets or laptops (**targets**), as well as static devices, such as WiFi routers or access points (**anchors**). Positioning systems can be built on top of WiFi without any modifications to the existing communication standards.

WiFi is the most interesting of the radio-based communication technologies

for adaptation into positioning systems because it is popular (WiFi is usually turned on, compared to Bluetooth which is offline by default) and has a small transmission range compared to GSM or 4G, resulting in higher positioning accuracy. Another advantage is that unlike GSM and 4G, WiFi access points are mass products. This makes WiFi devices cheap and positioning systems based on WiFi affordable.

WiFi self-positioning is widely used as a low-energy, low-accuracy replacement for GPS [15]. It takes advantage of WiFi access points, acting as static anchors, which are uniquely identified (to some degree) and have been previously mapped using wardriving[3] [16]. Smartphones having the Android and iOS operating systems use WiFi for self-positioning [17, 18] taking advantage of crowd-sourced [19] maps with the positions of WiFi access points (anchors).

The positional accuracy of WiFi self-positioning leads to applications such as flock detection [20] (detection of groups of people walking together). More complex applications based on WiFi self-positioning reveal the potential of the positioning data for conducting movement and social analysis [21]. However, in all cases of self-positioning the data is limited as few users want to contribute. The two works (flock detection and social analytics) are based on studies of tens of individuals.

Gathering long-term positioning data over large areas for many individuals has proven to offer interesting results. These data sets have been analyzed in order to extract complex information such as life-pattern analysis [22], social interactions [23] or facility utilization [24].

The potential of WiFi remote positioning has made it popular and a large body of research has appeared based on the technology. The expectations are large, with researchers making claims of the ability of the technology to be used for crowd-dynamics monitoring and modeling, as early as 2010 [25].

The research reported in this thesis focuses on exploring the potential and limits of WiFi remote positioning for crowd-dynamics monitoring. We know that interesting results can be obtained for long time frames, so in order to truly test the limits of WiFi remote positioning we concentrate on determining what information can be extracted from data pertaining to small time frames (days), over small areas (city center or campus) for many individuals.

[3]Wardriving is the process of driving around a city, recording the GPS position where WiFi access points are detected. It builds a map of WiFi access points

CHAPTER 2

Positioning and WiFi remote-positioning systems

There is a large variety of positioning systems. They make use of various modalities of identifying a targets location, going from visual data to electromagnetic signals. Each has advantages, disadvantages, and different use cases.

New developments bring positioning systems that can be used to monitor large crowds. This opens the way for smart-city applications, better urban planning, improved safety, marketing, etc. In this chapter we explore the available positioning systems and explain why we chose WiFi remote positioning as the one best suited for crowd-dynamics monitoring.

WiFi remote positioning is already popular for monitoring. We studied many of the projects that utilized the technology and implemented our own systems. Using these systems, we gathered multiple data sets. Data sets used to better understand the capabilities of this technology.

2.1 Contributions

Crowd-dynamics modeling requires many traces obtained by monitoring positions of many individuals. Traces can be built given a list of timestamped positions. There are many options when it comes to the choice of positioning systems. In this chapter **we offer a survey of the most popular positioning systems and describe the properties of each**.

One of the most important, recently developed, positioning systems is based on WiFi. **We show how WiFi remote-positioning systems compare with others** and **we describe our implementation of a WiFi remote-positioning system**, its components and explain how the data-gathering process works, what are its advantages and disadvantages. **We also present the notation that we**

use throughout the thesis.

Although there is now significant literature for WiFi remote positioning, as to our knowledge, **no other work has described many of the details that need to be considered for WiFi crowd-dynamics monitoring system implementations**. Among those details we consider factors such as the choice of channel. WiFi uses multiple frequencies and the hardware commonly used for WiFi platforms can listen only on one frequency at a given time.

Using the crowd-dynamics monitoring system that we described (as well as similar ones) **we perform five data-gathering experiments**. These experiments span over five years, multiple cities and represent different contexts. They total in data representing a month of positions for hundreds of thousands of individuals. The data obtained from these experiments is used in the other chapters in order to gain a deeper understanding on the potential and limitations of using WiFi remote positioning. ***Preliminary analysis of the raw data*** offers some interesting insight on the capabilities and limitations of these systems. As will be discussed at various points, we have taken care that the privacy of individuals has been preserved. Secure encryption techniques have been applied in addition to providing information and opt-out options where appropriate. Furthermore, data has been used only for this research, namely for the purpose of investigating the usability of WiFi remote positioning. The data sets are destroyed when the thesis is published.

2.2 Survey of popular positioning systems

For the purpose of this thesis we need to identify positioning technologies that can be used for crowd-dynamics monitoring. By having time-stamped positions of people, we may be able to trace their movements and multiple movements can represent crowd dynamics. Such a positioning system needs to function with many targets (people) and cover large areas while offering details about their movement and position. Another important benefit to consider would be the cost of such a system.

WiFi remote positioning is the positioning technology on which this thesis focuses. This is because WiFi remote positioning fits all the criteria required by a crowd-dynamics monitoring system and is currently the best at doing so. The goal of this section is to present all other positioning technologies and motivate our choice for WiFi remote positioning.

Today, the term “positioning” refers to an extensive set of processes, across different fields: it is used in psychology and sociology [26] representing how

people compare themselves to others; in marketing [27], showing what companies want people to feel about their brands; in physics where it can represent placement of elements as small as individual atoms [28]; in medicine, where it is used to determine the location of cancer cells [29] or at an even smaller level, at determining the location of chromosomes inside the nucleus of cells [30]; and many others.

Current technology permits us to measure people's position at very accurate levels (millimeters or centimeters) but these solutions are designed to work for individuals inside well-controlled environments. A few of these systems are used for: motion capture [31], computer-assisted surgery [32] or entertainment [33, 34] (Wii and Kinect).

Even when we concentrate on determining the position of humans as they move around a city, there is still a large variety of well-established technologies that can be used. Each of them has different advantages and different scopes and they are not easily interchangeable. The most popular large object-positioning systems use visual sources, sonic, electromagnetic signals, or come as extensions of established communication protocols.

2.2.1 Visual systems

Determining the position of people can be done by using visual sources. Visual positioning systems consist of video cameras that record continuous feeds. Given the position of a camera, the video stream can be processed in order to extract the position of each individual that is recorded. Camera systems are common, especially in residences, where they are used to offer security. With the same purpose they have been used on city scale, like is the case for London's CCTV [35]. More recently, they were used to make measurements of crowds. The Advisor [36] system, designed for public transport, can offer information on crowd densities to help prevent overcrowding or even identify potentially dangerous situations.

Positioning systems based on video streams have an important advantage of being simple to validate. Errors in data extraction can be corrected by manually verifying video logs. However, this is a timely and costly procedure.

The main deterrent from choosing visual-based positioning systems is the cost [37]. The cost is given both by the camera itself and by the support systems required to stream and process the visual data. These costs are going down with advances in computer vision [38]. Work that had to be done exclusively by humans is now taken over by software. But, even with these advancements

we are still far from achieving the requirements for affordable large-scale visual positioning system.

Visual systems also bring some important privacy concerns. Being filmed constantly raises ethical questions because people can easily be identified. This issue has been addressed in recent research by automatically hiding people behind silhouettes [39], but it is not yet clear how much of the privacy concerns this solution manages to address or how performant it is for dealing with large crowds.

2.2.2 Radar/Sonar systems

Sound navigation and ranging (Sonar) [40], and radio detection and ranging (Radar) [41] work by recording sound or electromagnetic waves, respectively, and determining the distance between the radar/sonar device (the anchor) and the targets. The initial waves can be generated by the target or the environment, in which case the systems would be passive (an example would be ASDIC [42]), or they can be generated by the radar/sonar device, in which case the target would reflect the waves.

Initial radar/sonar systems could determine the position of only one target but they have been improved in order to support multiple targets [43, 44]. However, the number of targets remains limited and it is not clear if these systems can reliably determine the position of individuals in large crowds. Other improvements have increased the positioning accuracy making them usable for indoor environments, like the Bat Ultrasonic Location System [45], but outdoor performance and scalability have not yet been achieved.

Similar systems have tried to use this technique for dealing with crowds. This is the case of Electronic Frog Eye [46], which uses channel state information from WiFi signals to determine the number of individuals in crowds. Although not exactly radar, the principle is similar, information from the recorded signal is used to determine how many people are inside a room. This system measures only the number of people and not everyone's trajectory or position and requires some careful calibration. This makes it unfeasible for measuring crowd dynamics. Furthermore, these systems require a, possible extensive, processing phase before the information can be extracted.

2.2.3 Systems with active anchors and target

Systems with active anchors and target offer high positioning accuracy at a high frequency and scale to large numbers of people. These systems make use of

electromagnetic signals that are transmitted by one of the entities and received by the other. Many versions exist:

- **Global navigation satellite system (GNSS)** [47], with the most popular implementation being the Global Positioning System (GPS) [2] is the most commonly used method to identify positions. It works by having a network of satellites, with known positions, continuously broadcasting signals. The signals are received and analyzed by a device small enough to be carried by an individual. By comparing the differences between multiple signals, the device can calculate its position relative to the satellites and determine its longitude and latitude. The main advantage is that GNSS systems work from anywhere in the world and the accuracy is in the order of meters.

- **WiFi or Cellular self-positioning** [15] is used by the most popular smartphone operating systems, iOS and Android. They represent an energy efficient low-accuracy positioning technique. Most WiFi routers transmit Beacon frames in order to signal mobile devices that they are in range and the network is available. By using maps of WiFi router positions such as WiGLE [1], one can determine a mobile device's position by determining which WiFi routers are in range.

- **Active badges** [48] are proprietary devices that transmit beacons between them or to and from base stations. These beacons can be used to determine the location of the person carrying a badge. Because all elements, transmitters and receivers can be finely tuned, this system can offer high positioning accuracy. Furthermore, the badges work both indoor and outdoor. An important aspect of active badges is that they can be worn in such a way that the *direction* in which a person is facing can be reliably determined. The human body obstructs many transmissions, making the antenna of the badge act like a directional one. Using this feature, studies have been recently carried out that show the system is able to determine trajectories of visitors at an exhibition as well as at which exhibit they were looking [49].

The biggest disadvantage of all these systems is that they require the target to be directly involved in the positioning process. Although they are perfect for personal use, this makes them expensive to deploy and even unrealistic for

[1] WiGLE`https://wigle.net` (Accessed 17-May-2019)

large scales. People are not willing to carry new devices, and even when we use the ones they already have, the smartphones, they are reluctant to install new software that could be used to send the data to a centralized location. Furthermore, technologies such as GPS can consume considerable battery load and produce heat, making the user even more unwilling to participate in such a data-gathering process.

2.2.4 Remote positioning based on communication systems

Most of us carry smartphones. These devices have many features, including support for multiple communication protocols. They support GSM for voice communication, 4G for data, WiFi for data inside our homes or offices, Bluetooth for connecting to external devices and NFC for contact-based operations. Improvements and new protocols appear all the time: 5G is being released, WiFi is at 802.11ac, and Bluetooth at 5.1. For all these protocols, the smartphone can act as both a receiver and a transmitter of electromagnetic signals.

We can build positioning systems based on the signals transmitted by any of these protocols. This can be done based only on the signals that are already sent by our devices without adding any new transmission. The basic principle is simple. The protocols enable communication between a statically placed base station and the mobile device (be it a smartphone, laptop, tablet or otherwise). The base stations can act as our anchors, with known positions, while the mobile device represents the target. We use the signals to determine the position of the target relative to the anchors.

We know that if one device receives an uncorrupted transmission from another, the distance between the two devices is at most equal to the maximum transmission range for the given protocol. This is not exactly true, considering the transmission range does not have a fixed value and it is affected by many elements. To name a few, the transmission range can be shortened by obstacles or the weather and extended because of tunneling effects. Even so, we can approximate the position of one device to be equal to the position of the other with an accuracy of a distance close to the transmission range of the given protocol.

Determining the position can be done both at the target (self-positioning) or at the base stations (anchor). It is possible to improve the accuracy by receiving simultaneous signals from multiple anchors, or by receiving the signal from the target at multiple anchors. This means the target would be in the zone where the coverage areas of the anchors overlap. We can further improve the accuracy by making use of the strength of the received signals.

Self-positioning systems based on communication protocols have a higher frequency of recording positions compared to remote-positioning systems. This is because the access points do not have energy limitations and send signals at a relatively high frequency. However, self-positioning assumes the involvement of the target, meaning the system cannot scale to many targets.

Remote-positioning systems based on repurposed communication protocols easily scale to many individuals. However, having a remote-positioning platform assumes that the anchors have centralized control. The costs remain low because networks of access points can be reconfigured to act as sensors and deployment of new platforms incur only the cost of access points.

The popularity of smartphones and the wide use of communication protocols enables the development of positioning technologies that scale to previously unrealistic number of targets. Each of the communication protocols brings different properties to the resulting positioning data set. The main communication protocols that can take advantage of large-scale use in order to offer crowd-dynamics monitoring are the following:

- **Global system for mobile communications (GSM)** [50] is the standard used by almost all mobile phones for voice communication. Every time a call is made, or an SMS is sent from a phone, a record is kept by the company that offers the phone service. The records are used for billing purposes. These records contain the id of the phone, the time, as well as the id of the cell tower to which the phone was connected. These data sets are called "call detail records" (CDR).

  Using CDRs we can approximate the position of the phone to the position of the cell tower to which the phone was connected to. This offers a low positional accuracy because cell towers can transmit and receive signals for distances in the order of kilometers (as much as 35km). Furthermore, because records are generated depending only on user interaction the frequency at which data points are generated is low and varies depending on both the user and the time.

  With rare records of small positioning accuracy, traces based on call-detail records lack information on all places in which we do not make phone calls (e.g. shops, coffee places). The advantage is that they can easily achieve large scales. Service providers can serve millions of users and cover entire countries. This makes them ideal to study large-scale behaviors such as measuring seasonal patterns [51].

- **3G/4G** [52, 53] are the data transmission technologies used by mobile

phones for using cellular data. The access points are controlled by service providers. Similar to GSM, the service providers keep logs when a data transmission is being made. These logs can be used to extract positional data. The positioning accuracy is given by the range of the 3G/4G signal, which, although shorter than GSM, remains in the range of kilometers. Because usage of 3G/4G incur costs, smartphone applications and operating systems try to limit the usage. This means the frequency at which data points are added to the logs, and in turn the frequency of positions, is low.

- **5G** [54] is starting to be deployed. It functions at a range of hundreds of meters, making it comparable to WiFi as opposed to standard cellular technology. The protocol also makes use of beamforming, a technique of directing the signals. This could be used to further improve the positional accuracy. The small range gives it an important advantage when considering the choice of positioning system. Unfortunately, wide adoption of the protocol is still far in the future. This means that it will take some time before it becomes viable to use as a positioning system.

- **WiFi** [55] works at a range of about 100m. Its hardware is commercially available, both in the form of mobile devices and access points. Signals are transmitted in order to serve the requests of the user but also automatically, in the form of control frames. We will discuss WiFi in more detail as it is the focus of this thesis.

- **Bluetooth** [56] has a transmission range in the order of tens of meters, offering higher positioning accuracy compared to the other technologies [57], but requiring more sensors to be placed to cover an area. The cost of deploying more sensors can be significant and deter the usage of Bluetooth for such applications.

  Bluetooth is not as widely used as compared to WiFi. Although it is present in most smartphones, it is not enabled by default and requires peripheral devices (e.g. Bluetooth headphones or speakers) to be useful. Wearables that connect to smartphones may cause Bluetooth to be more popular. Not being popular, however, means that they generate less data, fewer transmissions, and in turn, a lower number of positions compared to WiFi.

Out of the communication protocols WiFi is the most promising for crowd-dynamics monitoring. It is widely used, with most of us carrying WiFi-enabled devices which could potentially be tracked. It offers a reasonable positional

accuracy for outdoor settings, of around 100m, with positions recorded at a possibly high frequency rate. The frequency of positions may be high because positions are recorded both when the target device is used and when it automatically transmits control signals. And all these benefits are given while remaining unintrusive, which allows platforms to scale to large number of people. WiFi is also the least expensive technology to use because the required hardware for anchors is mass produced.

2.3 WiFi remote-positioning system

Our work focuses on WiFi remote-positioning systems as they are currently the most promising technology for conducting crowd-dynamics monitoring and analysis because it has the potential for easily providing significant and relevant positioning data. With large amounts of positioning data, we can extract more information that can be used to model crowd dynamics.

The advantage of using WiFi remote positioning is that we do not need to have control over both the targets and the anchors because both already transmit signals. We need to modify only one of these components and make use of the signals transmitted by the other. The chosen component is modified so that it captures and records the electromagnetic signals and calculates the target's position based on them. Regardless of our choice, the components of WiFi systems are widely deployed.

Smartphones are ubiquitous and carried with us at all times. They are, however, difficult to modify. There are multiple variations in both hardware and software requiring a lot of work to make any system work for all smartphones. Furthermore, any modification can be done only with the cooperation of the owner.

Scaling to many targets can be done only if we have control over the anchors. Control means we can modify the software/hardware of the anchors. This way, we can build a WiFi remote positioning framework that has a small deployment cost (installing sensors) and scales to many targets (as many as fit in the area covered by the sensors). In some cases, the deployment cost can be lowered even more by converting existing WiFi access points to sensors. This implies minimal modifications to the software running on the access points.

A WiFi remote positioning system (where only the anchors need to be controlled) takes the form from Figure 2.1. It has the following components: the device carried by the **target** individual, which during normal operation sends WiFi frames to find and communicate with WiFi routers; specialized sensors

(**reference points or anchors**) that receive signals broadcast by the device in the form of WiFi frames; a server that gathers the positioning data.

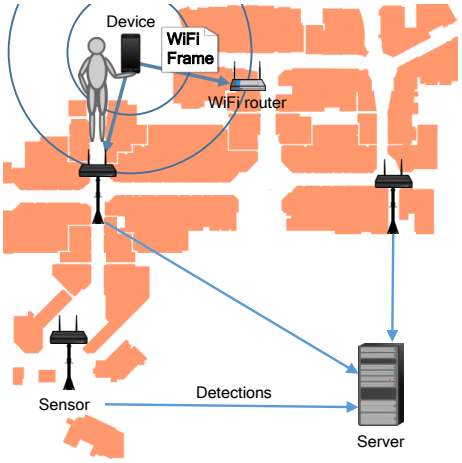


Figure 2.1: WiFi remote-positioning

To simplify the presentation, we use the term **device** to represent the target (individual and WiFi enabled gadget, smartphone, carried by the individual) and the term **sensor** to represent reference points.

The sensors are passive, they do not participate in the WiFi frame exchange. The frames are sent only between the device and the WiFi access point to which it's connected, or broadcast when the device is searching for a new network.

When a WiFi frame is received by one of the sensors a **detection** is generated. A detection contains: a **time stamp**, identifying the moment when the frame was received; the **sensor id**, a unique identifier given to each sensor, interchangeable with the geographical location of the sensor; a **device id**, uniquely identifying a device. Regarding the positioning of the device, and in turn the individuals represented by a detection: Assuming two or more sensors detect a device simultaneously, these detections can be combined to obtain higher accuracy positions. This is commonly what is referred to when we use the term "positioning". In practice, for our scenario of outdoor detections and crowded environments, we have discovered simultaneous detections are rare. When only one sensor records detections of a device, the location of the device can be approximated to that of the sensor. This is because of the limited WiFi range, making the detection range of a sensor limited. As each detection reveals

the position of the device as being near the sensor recording the detection, we consider throughout our work that detections provide positions and **recording detections is a form of positioning**.

WiFi remote-positioning systems can handle multiple targets due to the **device id**. The device id can be obtained by taking advantage of the specifications of the 802.11 protocols.

2.3.1 Using the 802.11 protocols

The 802.11 family of protocol standards defines the physical layer and medium access layer for wireless data communication. These layers are part of the TCP/IP stack [58] which is used for most data communication.

On the medium access layer, the standard defines frames as the communication entity. Frames represent structured data, which is sent to the physical layer, encoded and transmitted as electromagnetic signals. At the receiver, the signal is interpreted, and the frames are reconstructed. Whenever we discuss detecting of a device through WiFi, we mean receiving and recording WiFi frames.

Frames have a general format from which 39 frame types and sub-types are derived, as well as a few reserved ones. This format is presented in Figure 2.2. It is common that the first three addresses be present and represent the source address (SA), destination address (DA) and the basic service set identifier (BSSID - identifies a network) respectively. There are frames that do not contain all the fields. For instance, Clear To Send (CTS) frames, used to signal that there are no other transmissions taking place, do not have a source address. Table 2.1 contains a list with all frame types/sub-types that contain a source address.

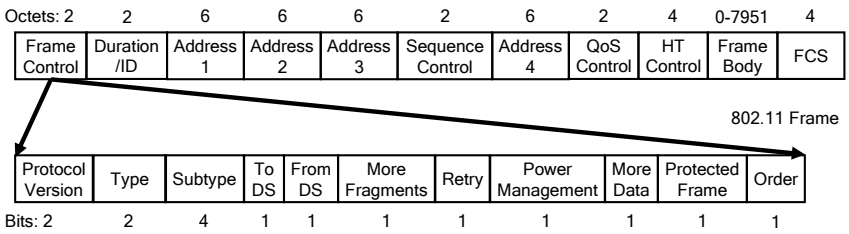


Figure 2.2: WiFi, 802.11 General Frame Format

Every device that uses WiFi has an address. When transmitting data this address is included as the source address in some of the frames. It is reffered

Table 2.1: 27 frame types/sub-types that contain a source address

Type	Sub-type		
Data	Data	Data+CF-ack	Data+CF-poll
	Data+CF-ack+CF-poll	Null	CF-ack
	CF-poll	CF-ack+CF-poll	QoS Data
	QoS Data+CF-ack	QoS Data+CF-poll	QoS Data+CF-ack+CF-poll
	QoS Null	QoS+CF-ack(no data)	QoS+CF-poll(no data)
Management	Association_Request	Reassociation_Request	Probe_Request
	ATIM	Disassociation	Authentication
	Deauthentication	Action	
Control	Block_Ack_Request	Block_Ack	PS_Poll
	RTS		

to as a MAC (media access control) address and is set by the device manufacturer. IANA [2] provides OUI (Organizationally Unique Identifier) numbers for hardware manufacturers for this purpose. The first 24-bits of the MAC address are set to be the OUI and the rest of the bits are set to a value decided by the manufacturer so that each device can be uniquely identified.

Even though the intention of the standard is to have unique MAC addresses for each device, this rule cannot be enforced. In most cases the MAC address can be changed through software. Changing the MAC address requires some technical skills and because of this, most people do not modify it. This means that although not guaranteed to be unique, it is common for the rule to be followed. More so, the standards cannot handle two devices in the same network having the same MAC address. As such we can implement systems with the assumption that MAC addresses will be unique.

Because we can assume the MAC address to be unique, and most frames contain the MAC address of the device set as the source address we can use the value of the source address as the **device id**, a unique identifier for the device. The device id can be used to correlate detections of a device across multiple sensors. Uniquely identifying a device makes it possible for WiFi remote-positioning systems to have multiple targets.

No available encryption can stop or interfere with WiFi remote positioning. When the connection is using security protocols such as WPA2 [59] only the frame body is encrypted. This means the source address which is part of the head and not part of the body of the frame, is always available to any listening equipment.

Some positioning systems can trigger false detections. This is not the case

[2] `http://www.iana.org/` (Accessed 08-Feb-2017)

for WiFi remote positioning. An example of a false detection would be a radar detecting a plane that is not there. This can be caused due to interference or noise. WiFi uses 2.4Ghz and 5Ghz frequency bands and encodes the data in a digital form, making its signals distinguishable from the noise found in nature. There is also a Cyclic Redundant Check (CRC) number in the FCS field of every frame. The CRC is used to identify transmission errors. If the CRC is missing or does not match the expected value, the frame is marked as malformed and can be dropped. All these features of the WiFi protocols guarantee that radio frequency signals cannot be interpreted to be a WiFi frame.

In order to communicate, WiFi devices need to be connected to a network. A WiFi network is generally built from several access points, each offering mobile devices the possibility to connect to. A large network can easily consist of a few thousand access points and allow for tens of thousands simultaneous connections. Because WiFi networks are limited in radius and devices are assumed to be mobile the standard describes two ways for a device to discover a network:

- An access point advertises its SSID (the name of the network) and network characteristics (eg. accepted bandwidths) by broadcasting *Beacon* frames. Devices listen for *Beacon* frames and start the connection process if the network is known or if the user requests it.
- Devices can actively scan for networks by broadcasting *Probe_Request* frames. This allows a device to connect to a network that does not transmit *Beacon* frames (a hidden network). It also enables faster network discovery because the device no longer has to wait for *Beacon* frames. Once the network is identified the connection process can start as usual.

Because of roaming, *Probe_Request* frames are normally sent even if a device is connected to a network or not. Roaming represents a mobile's device capability to connect to a different access point of the same network without causing an interruption in service. When a device is connected to a network, it may continue to send *Probe_Request* frames in search for an access point that can offer better connectivity, based on bandwidth or signal strength.

WiFi works on multiple frequencies. These frequencies are called channels. There are 14 channels in the 2.4 GHz frequency band, with more in the 5 GHz one. Communication inside a network, after the connection is established, is done on a fixed channel. If the sensor listens to a different channel than the one the network is on, it will not receive any communication frames (eg. *Data*), rendering connected devices invisible. It is common for access points to set the

channel in an automatic manner, allowing them to change it in time, but these changes are rare. *Probe_Requests* are the only frames that are sent to multiple channels, making them ideal for WiFi remote positioning.

The standard does not contain any specification on the channel to be used or how often to transmit *Probe_Request* or *Beacon* frames or how to listen for them. This is left to the manufacturer and in the case of smartphones these parameters are affected by multiple conditions such as the battery level or the screen status. A device that is low on battery tries to conserve it as much as possible and sets a low frequency at which to send *Probe_Request* frames. In our experiments, we found it common for *Probe_Request* frames to be sent with some regularity, usually about one per minute. This frequency is higher than the frequency of recordings for other large-scale positioning methods, such as the ones that make use of call detail records [1].

It is possible to have a sensor that functions as a WiFi access point. This is especially useful when an already deployed WiFi network with multiple access points is configured to have it conduct WiFi remote positioning. This works without disruptions to the normal functioning of the network. However, to continue functioning correctly, the sensors can listen on only one channel. The channel does not have to be the same for all sensors, but it cannot change over time without explicit intervention.

Commercial WiFi devices have an advertised communication range of 100m, without obstructions. This is the range at which frames should be correctly received. Beyond this range the signal strength decreases too much and the environment noise makes it impossible to reconstruct the frames correctly. The signals are affected by the environment making the distance and the shape of the area in which frames can be correctly received vary. Given the limited range of WiFi, when a sensor detects a device, we can approximate the position of the device to be that of the sensor. The approximated position would have a measurement error that could at most be in the range of the communication distance for WiFi (plus a margin to account for effects such as tunneling that can extend the distance).

There are many factors that make WiFi remote positioning systems interesting: WiFi is widely available and popular, making it cheap; the protocol uses unique addresses for each device, permitting us to find the positions of multiple devices (targets) simultaneously; the signal and frame specifications make false detections impossible; frames are sent with some regularity, in the order of minutes; the minimal positioning accuracy is of at most a couple of hundred meters.

2.3.2 WiFi remote-positioning system implementation

The basic concept of gathering WiFi frames to model crowd dynamics is simple. To receive WiFi frames all that is needed is a WiFi device. Many such devices are available, making the process easily accessible and cheap.

We have conducted multiple experiments using different hardware and software configurations. To offer a concrete example, we primarily use a platform consisting of sensors (Bluemark 1000 series) and a server to centralize and store the data. The Bluemark BM1000 sensor has 32 MB RAM, 8 MB flash and a 384 MHz CPU. The sensor uses a directional WiFi antenna with a gain of around 12 dBi and has a 4G module for communication with the server. It runs OpenWRT [3] as its operating system, together with an application we developed that sets the WiFi interface to monitor mode and records WiFi frames.

Most WiFi remote-positioning deployments use commodity hardware which does not have directional antennas. Using directional antennas would have the potential of increasing positioning accuracy by limiting the sensing area. Throughout our research we purposefully ignore the fact that antennas are directional in order to not have an unfair advantage compared to other deployments. We are aware that future technologies, such as 5G, will introduce beam steering [60] which will enable increased positioning accuracy, however we focus on existing and widely available technologies.

The sensor registers a **detection** whenever a frame is received. Not every received frame can be used to record a detection. For instance, frames without a source address cannot be used because they cannot be matched to a device. A detection has the source address hashed and saved in SQL text format, compressed and periodically sent to the server. At the server, the files are decompressed and set for long-term storage and analysis. The sensors are synchronized using the Network Time Protocol (NTP) [61] and they reboot daily at 5AM. This last step is meant to minimize errors.

The application we developed for the sensors uses the libpcap library [4] to access the WiFi interface and receive WiFi frames as well as an MD5 [62] library to hash the device identifiers (MAC addresses). Hashing is an important step because it is possible to link the MAC address to a device [63], while the secure hashing prevents this. We aim to preserve the privacy of individuals being monitored to the best extend possible. Access to the data sets is limited to only authorized people.

As for the sensor itself, it is placed in a simple, small box, along with a power

[3] `https://openwrt.org/` (Accessed 16-Feb-2018)
[4] `http://www.tcpdump.org/` (Accessed 16-Feb-2018)

Figure 2.3: Installed Bluemark 1000 sensor

source, as can be seen in Figure 2.3. The picture represents a sensor installed and used during one of our experiments. The sensors are generally placed on streetlight poles.

2.3.3 Notations

We use the following notations for WiFi remote-positioning data sets.

S - the set of sensors (anchors) $\{s_1, ...s_N\}$ labeled with sensor identifiers, interchangeable with the sensor position.

D - the set of devices (targets) $\{d_1, ...d_M\}$ labeled with the device identifier.

t - time, measured in seconds.

λ - is a detection, a tuple of sensor id, device id and time $\langle s, d, t \rangle$.

λ^S - is the sensor that recorded the detection λ.

λ^D - is the device identifier recorded for detection λ.

λ^T - is the time at which the detection λ was recorded.

Λ - is the set of detections $\{\lambda_1, ...\lambda_R\}$.

$\Lambda[d]$ - is the set of detections $\{\lambda_1, ...\lambda_R\}$ belonging to device d.

$\Lambda[s]$ - is the set of detections $\{\lambda_1, ...\lambda_R\}$ belonging to sensor s.

$\overline{\Lambda} = (\lambda_1, ...\lambda_R)$ - represents a sequence of detections so that they are ordered by device, time and sensor.

$\overline{\Lambda[d]} = (\lambda_1 ... \lambda_P)$ - represents a sequence of consecutive detections of device d. It is a subset of $\overline{\Lambda}$ in which $\forall i; \lambda_i^D = d$. $\overline{\Lambda[d]}$ represents the trace of a device d.

$\overline{\Lambda[s]} = (\lambda_1 ... \lambda_S)$ - represents a sequence of consecutive detections at sensor s. It is a subset of $\overline{\Lambda}$ in which $\forall i; \lambda_i^S = s$.

2.3.4 Two sensors experiment - choosing the channel

The WiFi protocol enables the use of one of several channels to transmit data. If a frame is sent on one channel, the listener must have the same channel set to be able to correctly receive it (frames on adjacent channels can be received but it is less likely). Furthermore, a listener cannot receive frames on different channels simultaneously (an exception can be considered for similar channel frequencies). Considering this restriction, a WiFi remote-positioning system needs to consider which channels to have the sensors listen on. There are two popular options: choose one channel and listen only on that one or change the channel at a regular interval. Changing the channel is called channel hopping.

We conducted a small-scale experiment to determine if there are differences between what two WiFi sensors placed next to each other receive. During the experiment we made a special consideration on channel hopping. We had one sensor listening on one channel and the other listening on the same channel, channel hopping and listening on a different channel. We deployed two sensors in a laboratory room and had them receive frames for about three hours. These sensors were made by the same manufacturer, had the same antenna, and were placed about 50 cm apart.

The first sensor is configured to listen on channel 3. The second sensor is configured to listen on channel 3 for one hour, do channel hopping for the next hour and for another hour listen on channel 8. Using this partitioning, we have a period when both sensors listen on the same channel, a period in which one remains on one channel while the other does channel hopping and a period when they listen on different channels.

During the experiment we had eight WiFi devices in the laboratory. The list of devices is presented in Table 2.2. As can be observed, the devices vary by operating system and type, offering a broad view of possible devices. These devices moved very little during the experiment and were not removed from the laboratory room.

We recorded the time at which a frame was received from the eight devices on each of the two sensors. The results are presented in Figure 2.4. The gray area is the area in which one sensor was doing channel hopping. The white

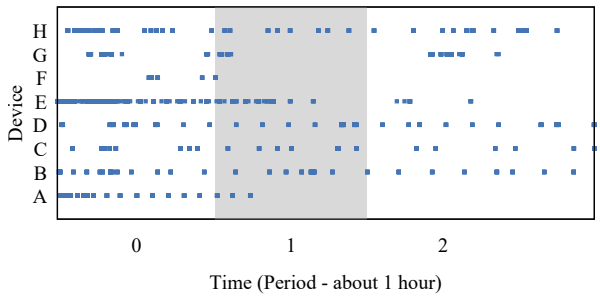
Table 2.2: Two Sensors Experiment - Devices

Name	Device	Operating System	Type
A	Nvidia Shield	Android	Tablet
B	Samsung Galaxy S7	Android	Smartphone
C	Lenovo Y50	Windows 10	Laptop
D	Mac macbook pro	iOS8	Laptop
E	Nexus 7	Android	Tablet
F	Acer Aspire E15	Windows 10	Laptop
G	Nokia 6	Android	Smartphone
H	Nokia Lumia 625	Windows Phone	Smartphone

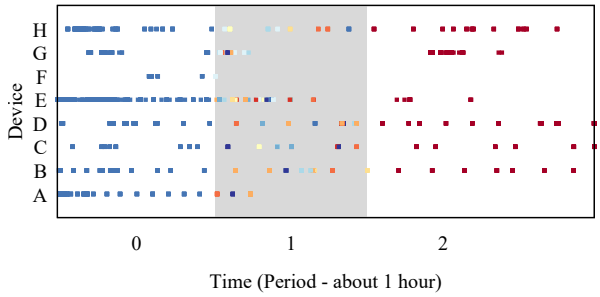
area on the left represents the period when both sensors listened to the same channel and the one to the right when the sensors listened on different channels. The different colors of the dots represent the channel on which a frame was recorded.

As can be observed there is little difference between the frames captured by the first sensor (Figure 2.4a) or the second one (Figure 2.4b), regardless of the period. In Figure 2.4c we show the difference between the two sets of detections, more specifically the difference between the union and the intersection of the two sets of detections. The only noticeable differences are for device G, but for that case the differences appear in all three periods, meaning the source of the differences is not the choice of channel.

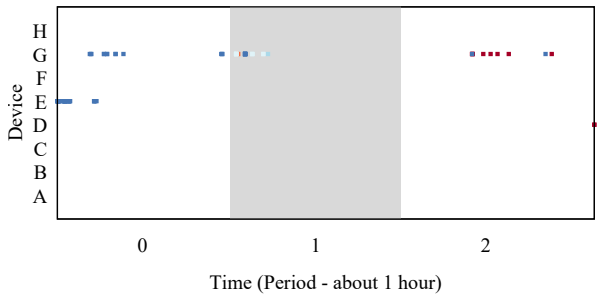
Based on our experiment we can only conclude that there is no significant difference between doing channel hopping or listening on a single channel. It is not clear if this is always the case. If a device transmits only data frames on a certain channel and does not transmit control frames, during this period the chances to detect the device are smaller if we hop channels and zero if we listen on a completely different channel. When we consider all frames received from all devices (devices that are not in the laboratory and can move without any control), we do notice a significant difference between the received frames, however the difference is the same regardless of the use of channel hopping. We selected 50 devices for which we display the difference in Figure 2.5 in order to offer an idea on this behavior. We go into more detail concerning these differences in the next chapter. Because for all devices the differences appear in the three periods, we conclude that the differences can be explained by the fact that different antennas have different chances of receiving low-quality signals correctly. This further confirms our analysis of channel hopping.



(a) Sensor 1 (Channel 3)



(b) Sensor 2 (Channel 3 - Channel Hopping - Channel 8)



(c) $\cup \setminus \cap$ of Detections at Sensors 1 and 2

Figure 2.4: Two Sensors Experiment - Channel Hopping. Data captured from two sensors and the difference between them. (Each channel has a different color.)

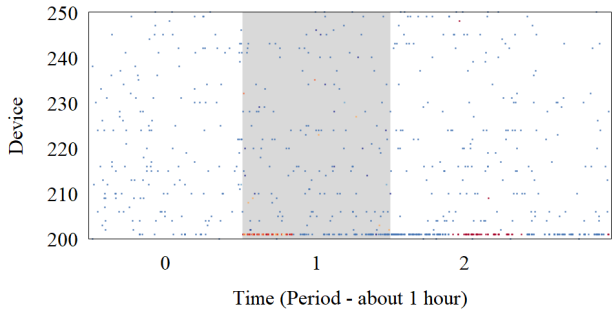


Figure 2.5: $\cup \setminus \cap$ of Detections at Sensors 1 and 2. Sample of devices

2.4 WiFi remote-positioning use cases

WiFi remote-positioning is now a popular technique for gathering data on people's location. A lot of research has been conducted that showcases the technology and its potential and many generic WiFi remote-positioning platforms have been described throughout the literature: Wombat [64], WiFiPi [65], Probr [66], WaP [67], SenseFlow [68], HABITS [69], Freecount [70].

Estimating crowd density is at the core of many applications that require positioning data. WiFi remote-positioning experiments that show crowd density measurements have been conducted given a variety of scenarios and contexts:

- In our own work [71], we showed how WiFi can be used to observe how the density of people changes throughout the day for a city center during a music festival.
- Labs, public exhibitions, lecture classes [72].
- Shopping malls [73].
- Motor show exhibition [74], where the authors confirmed the results by manually annotating visual data and using it as ground truth. The authors report that crowd density can be estimated with as little as 20% error.
- Lab class, footbridge, station ticket office and subway [75].
- Office space [76], where the authors used measurements of CO2 levels inside the room to confirm the results.

Determining the size of a crowd is only one initial step and one of many applications for crowd-dynamics monitoring for the scale and accuracy offered by WiFi remote positioning. A large variety of applications has been recently explored and studied. Here we present just a few examples:

- **Safety and security**: It has been shown that it is possible to monitor crowds using WiFi remote positioning and detect anomalous behavior through the use of outlier detection [77]. The authors developed an algorithm that finds differences from the normal behavior of a crowd and displays the last location of the individual having the anomalous behavior.

  An interesting application that can serve for disaster scenarios is the placement of WiFi sensors outside the building to determine the number of people inside [78]. The solution differentiates between devices behind and in front of the wall based on inter-event time differences added by the wall obstruction. The system has been proven to work for up to 20 people.

  Similarly, search and rescue can be conducted using a drone fitted with a WiFi sensor [79]. The drone can discover where people are trapped. The assumption being that individuals would have their WiFi devices with them yet be unable to utilize them.

- **Facility planning**: It is possible to use WiFi remote-positioning data to better understand the behavior of people with regards to the facilities they use. An example of this is the understanding of how people use the entrances of a hospital [80] to better improve the flow.

- **Tourism analysis**: The authors of Beanstalk [81] provided WiFi sensors to 60 locations on Madeira Islands in order to gather data for tourists. They showed that tourists can be differentiated from locals and that they can infer tourist behavior and trip itineraries as well as detection of meaningful events.

- **Traffic monitoring**: It has been showed that WiFi remote positioning can be used to monitor vehicular traffic [82]. The authors mentioned that their measurements managed to detect about one fifth of cars part of the traffic.

- **Mass transit management**: Unlike traffic monitoring, mass transit management focuses on understanding the usage of mass transportation systems (buses, trains). For their project, the authors [83] placed a sensor inside a bus that counted when people came on and when they left the bus. They confirmed the results by manually counting people that entered

and left the bus. Similar measurements were conducted in Aalborg [84]. As well as for the shuttle bus for Thammasat University, Thailand [85].

WiFi remote-positioning has been successfully used for rail transportation like in the case of the Melbourne experiment [86].

- **Social science experiments**: The project of [87] showed that encounters between individuals can be detected and this information can possibly be extended to determining friendships and automatically building social graphs. The experiment was conducted inside an open office room. Being inside the positioning data is more accurate than the one presented in our work. We show more about this in Chapter 3. The advantage of indoor measurements is made even more obvious considering it is possible to extract social relationships and interactions [88] by having just one sensor placed indoor, assuming enough time is made available to gather enough data.

  Multiple works [89, 90] show it may be possible to extract friendship information from the data advertised when devices are scanning. They showcased an experiment where they collected 6 months of data using a mobile sensor. This requires both a large time frame and access to data that we consider to be privacy sensitive and are not willing to record or utilize.

- **Classification**: There are multiple types of classifications that can be made based on WiFi remote-positioning data. The authors [91] showed it is possible to use WiFi to identify people's role, at least given simulated data. Their system uses expert knowledge and WiFi remote-positioning data to label individuals based on their roles (student, worker, cleaner).

  Devices can be classified as either laptops or smartphones [92] based on the pattern of detection.

  More interestingly, buildings can be classified [93] by discovering how often people visit them and with which regularity. The same can be done for public spaces [94]. The potential here being that this system of ranking locations can provide more accurate recommendations, immune to interference in the form of fake reviews that are abundant in crowd-sourced recommendation systems [95].

- **Business analytics**: Using real-time density estimates has marketing and resource-planning applications. It is possible to use density in order to

understand traffic flows inside shops. For example, this data has been used to measure the effectiveness of promotions [96].

2.5 Data-gathering experiments

We have conducted multiple data-gathering experiments spanning different years and locations. For three of the experiments we gather the data using the WiFi remote-positioning platform described in Section 2.3. The first experiment used a variation of the same platform. And the final experiment made use of an existing WiFi network. We modified the access point software so that it records detections and centralizes them similar to how the platform we describe does, but we also logged all associations to the WiFi network.

2.5.1 Privacy and ethical considerations

The ethical aspects on the privacy of individuals are an extremely sensitive topic. This topic has gone through extensive debate in recent years and more and more people consider it an issue. These issues have more recently taken the form of laws and regulations, such as General Data Protection Regulation (GDPR). Although this work focuses on analyzing crowd dynamics, which describe human movements on the scale of large groups, and as such obfuscating information on the individual, it does make use of positioning data at individual level. Because of this, we make the following considerations:

To respect the privacy of all individuals for which we record detections, the data is anonymized by applying a hash function on the MAC addresses along with a salt. The salt is not known by us, ensuring it is impossible for us to determine the detections of a specific individual assuming we had access to his MAC address.

From a WiFi frame it is possible to extract more data, such as: known SSIDs, supported transmission rates and others. This information has been previously used to identify a device [97] and it was shown that this data can be used to extrapolate more information on the device owner such as the country where she is from [98]. Because of these concerns, we only record the device identifier along with an identifier for the sensor and the time stamp of the moment the detection has been made. We make an exception of this rule for some experiments in which we collect other pertinent data such as sequence numbers. We specifically mention when this is the case.

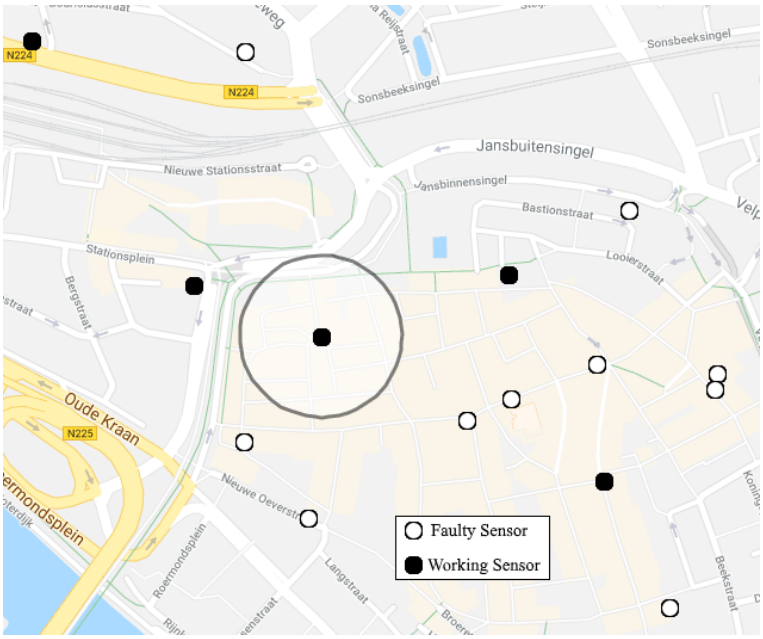


Figure 2.6: Position of Arnhem Sensors 2014 (the is an 100m visual guide)

To further ensure the privacy of the individuals being monitored we make sure our experiments cover only a small window of time, use different salts for different experiments, limit the access to the data to a few authorized individuals and only keep the encrypted version of the MAC addresses. As a separate project, we are looking into data-driven privacy enhancement techniques. This separate research is yet to be published.

2.5.2 Arnhem experiment

The first experiment we conducted was in the city of Arnhem, The Netherlands, in 2014. We deployed multiple sensors in the city during an event called the "World Living Statues Festival".[5] The event took place on the 28th of September and puts together participants taking the roles of living statues with visitors that admire their performance. The participants are given fixed positions on a

[5] `http://www.worldlivingstatues.nl/` (Accessed 16-Feb-2018)

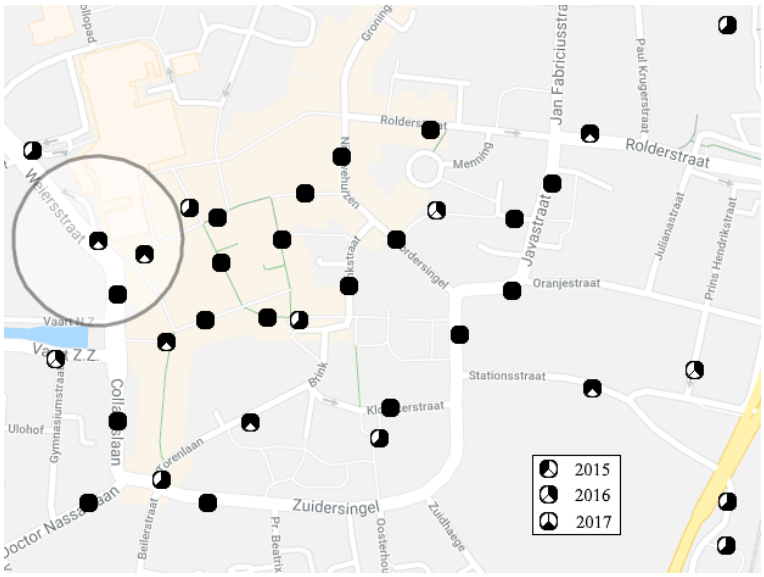


Figure 2.7: Position of Assen Sensors in 2015, 2016 and 2017 (The circle is a 100m visual guide. The tags represent sensor positions. Black parts of a tag show if the sensor was used in a specific year, according to the legend.)

path going through the city center. The visitors are advised to follow the path to observe all performances.

For this experiment we deployed 15 sensors. Because of a software issue only 5 of them correctly recorded detections. The other 10 had a faulty implementation which didn't properly record the device id. Although density of detections remains correct for all sensors, only data from 5 sensors can be used to build traces. The locations of the sensors can be observed in Figure 2.6.

2.5.3 Assen experiments

Once a year, the city of Assen, The Netherlands, is one of the hosts for the motorcycle grand prix [6]. The town organizes the TT Festival [7] in the days before the race. This festival offers a variety of activities: multiple stages for

[6] `http://www.motogp.com/en/event/Netherlands` (Accessed 17-Feb-2018)
[7] `https://www.ttfestival.nl/` (Accessed 17-Feb-2018)

music with dozens of performers, spread through the city center; a Ferris wheel; motorcycle stunts; a small amusement park; many restaurants, street-food vendors; camping area; and the TT Nightride event, where people are invited to drive their own motorcycle on a path through the city. The festival and races bring more than a hundred thousand visitors to the city. During the festival the city center is open only to pedestrians. This and the variety in activities generates a lot of pedestrian movement.

We conducted three experiments in the city of Assen, in the years 2015, 2016 and 2017. Our sensors were installed throughout the city center, as can be observed in Figure 2.7. The number of sensors used was different over the years as were some of the stage placements, but we tried to keep most sensors in the same positions. The black part of the marker indicates in what year the sensor location was used. There are four exceptions, of sensors placed further from the city center, near a camp and the racetrack. The sensors in the city center covered all the music stages. We recorded positioning data in the days during and around the festival.

2.5.4 Twente experiments

Our last experiment was conducted in Enschede, The Netherlands, at the University of Twente campus. This is our largest and most complex experiment. The sensors were the WiFi access points offering access to the Eduroam WiFi network. This enables us to record both Probe Request frames and connection status of devices.

2.5.5 Experiments summary

The experiments we conducted vary in the number of sensors used, as well as the area and activities that took place during the data gathering process. The statistics on each experiment can be observed in Table 2.3.

Our experiments contained test periods and sensors that were used only for testing. To clean the data sets we apply the following filters:

Duplicate detection filter - Because frames are sent at frequencies much higher than one per second and the time stamp of detections have a resolution at the level of seconds, it is possible to have multiple detections of the same device at the same time stamp and at the same sensor. We keep only one copy of such detections.

Interest time filter - It is common for us to test the system with a few sensors before a big event such as a festival. This data would skew our results, so we

Table 2.3: Data gathering experiments statistics

Location	Arnhem	Assen	Assen	Assen	Twente
Year	2014	2015	2016	2017	2018
Event	Festival	Festival	Festival	Festival	College
Sensors	15 (5 ok)	27	40	30	47
Raw Detections	2,373,494	15,135,611	36,331,241	26,414,742	16,541,927
Device ids	32,570	247,596	2,072,438	176,888	2,395,085
Estimate Devices	1,596	47,907	72,794	55,156	36,536
Frames	All	PReq	PReq	PReq	PReq (+ Conn)
Duration (days)	1	8	8	12	3
Channel hopping	no	yes	yes	yes	no
Start date	28-Sep	22-Jun	20-Jun	20-Jun	22-Apr

must keep detections only inside an interest time frame.

Interest sensors filter - Some sensors are used only for testing. We remove these from the final data sets.

The numbers in the table are representative of the data set, after these three filters are applied.

The number of device ids recorded varies from one experiment to the other. In the case of Assen 2016, the number of device identifiers goes over 2.000.000. This is impossible considering the city center can sustain at most a few hundred thousand people. The device ids are salted, hashed MAC addresses.

The large number of device ids in our data sets are caused by devices that change their MAC address, making each such device appear as multiple, different devices. This is the case when smartphones make use of MAC address randomization. Although there are techniques to go around this mechanism, like the ones presented in [97], these techniques represent a violation of privacy which we are not willing to make.

We found, based on the Assen 2015 data set, for which we record OUI values for every device, that many device ids having random OUI values (OUIs that do not appear on the public list[8]) have only one detection, and most have under 5 detections. This leads us to conclude that if we remove all detections for device ids that have a small number of detections, we clean the data set of these random MAC addresses.

To obtain a realistic number of devices we select the devices that have more than 39 detections. In Chapter 6 we find that if we only keep the devices ids for which we have more than 39 detections we remove most of the device identifiers based on random MAC addresses and are left with devices for which the data

[8] `http://standards-oui.ieee.org/oui.txt` (Accessed 06-Mar-2018)

is representative. In Table 2.3 we added the estimated number of devices and this values better match our expectations.

We investigated other possible causes for the large number of device ids detected in Assen 2016. We checked the distribution of device ids over time, over the sensors and the rate of change of device ids over time. We found no anomalies that would indicate a systematic error.

During our experiments we also considered different frames that are captured and recorded as detections. For the first experiment, in Arnhem, we considered all frames that had a source address. For the Assen experiments we considered that Probe Request frames are best suited because they are not affected by the network usage on the device.

For our last experiment, in Twente, we recorded both Probe Requests and logged the connection status of devices that use the Eduroam WiFi network at the university. Connections can be logged in existing networks without requiring specialized software, making it even more accessible to gather positioning data using this method. For this last experiment we were more aggressive regarding privacy and used different hash salts for every day of the experiment. This means that the number of estimated devices cannot be directly compared to the values from the other experiments because a device that is detected in multiple days is recorded with multiple device ids.

2.6 First glimpse of WiFi remote-positioning lackings

Our experiments have produced large amounts of data. We have gathered tens of millions of detections for hundreds of thousands of devices. They have been conducted in different years, under different circumstances, yet main characteristics remain similar. We do not add Arhnem to this comparison as the data from Arnhem come from only 5 sensors. As such, it is not sufficient or representative of what we consider normal WiFi remote-positioning data.

The detections that we have are not equally split between the devices. While some devices have thousands of detections or more, most of them have very few. Even after removing all devices for which we have less than 39 detections from our data set, more than 60% of devices have less than 200 detections. And this is true regardless of which data set we investigate. Figure 2.8 represents a histogram for the number of detections per device, for each of our four main data sets. The histogram shows percentages, as such, the results are normalized

by dividing the value for each interval of the number of detections per device, to the total number of detections for the data set. We apply the same technique to the other histograms in this subsection.

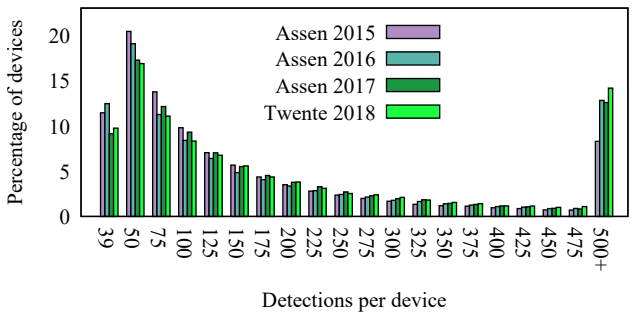


Figure 2.8: Histogram of number of detections per device

The histogram shows that most devices have few detections, but we also have devices that we detect hundreds of thousands of times. Unfortunately, devices with many detections tend to be static and do not offer any movement data. Static devices are continuously detected because they are plugged in and do not have a need to conserve energy. In these circumstances, devices can continuously scan for better connections and they broadcast probe requests constantly.

Static devices cannot be used for crowd-dynamics analysis because they do not add any information about movement or flows. However, we cannot simply remove all devices that have more detections than a given threshold. Consider a laptop that is plugged in at the office and at home. The laptop is mobile, and it may contribute information to crowd flows.

The frequency at which we detect devices varies from device to device and over time. It is possible to have two devices with hundreds of detections, one to be detected for a few minutes, and the other to be detected throughout the day. In order to have a realistic representation of how much time we detect a device we have grouped detections in slots of five minutes. We then add up the five-minute intervals to determine for how long we detect devices. We chose the interval size to be five minutes because it is small, yet large enough for a person to move out of the detection area of a sensor. Figure 2.9 is a histogram showing the distribution of devices based on the total duration for which each device is detected. Devices with fewer than 39 detections were removed prior

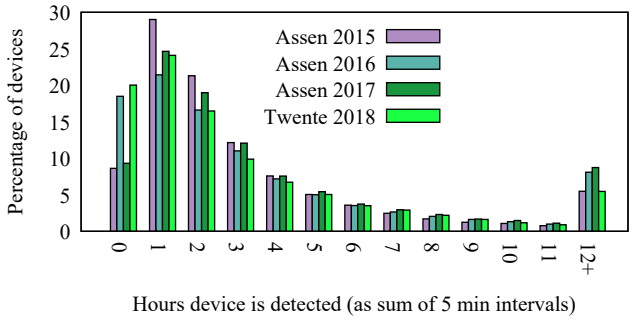


Figure 2.9: Histogram of total detection time per device

to building the histogram.

The distribution of devices based on the total detection time is similar to the distribution of devices based on the number of detections. Most devices are detected for few hours in total. The total detection time is different from the time difference between the last and first detection of a device. A device could be detected in the morning and in the evening but if it has few detections, the total detection time would be small. The total detection time is more representative because we have no information about what happens to a person or a device when it is not detected. It could be outside the detection area, or the device can be offline, or we do not detect it because of interference.

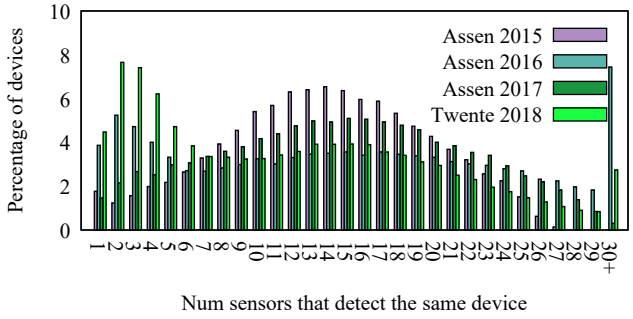
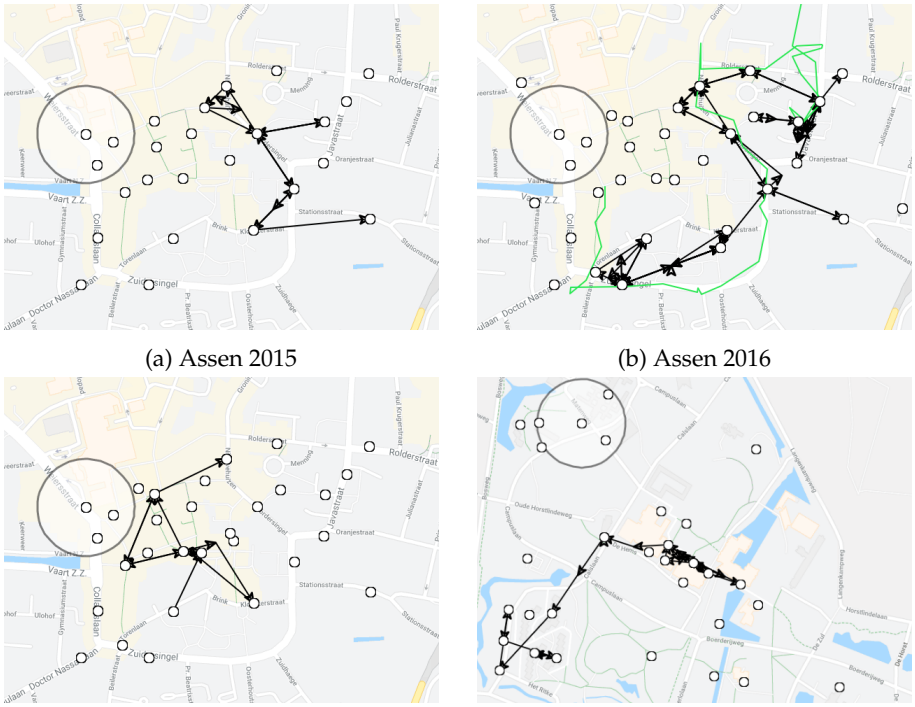


Figure 2.10: Histogram of number of sensors per device

Considering the large number of devices with few detections, we do observe

a lot of mobility. Figure 2.10 is a histogram showing the distribution of devices based on the number of sensors that detects the same device. There is a significant portion of devices detected by few sensors, meaning they are static. But most devices are detected by multiple sensors, with a calculated mean of 14, 14, 15 and 12 sensors detecting the same device.



(a) Assen 2015

(b) Assen 2016

(c) Assen 2017

(d) Twente 2018

Figure 2.11: Sample traces - each represents 1 hour of data (100m visual guide circle) Raw data is difficult to interpret

Having detected a device at different sensors is not enough to describe movements. If the sensors are close to each other, it is impossible to differentiate between a movement and a device being detected at different times by two sensors that are in range of it. When we use the raw detections to trace the movements of a device it is common for the device to appear to be moving in circles.

We extracted one example from each of our four main data sets to exemplify this circular behavior. The traces for these four devices are shown in Figure 2.11. For each device we have traced its movements for an interval of one hour. The white marks represent the placement of our sensors while the black arrows represent a movement. If a device is detected simultaneously at two or more sensors, we draw the arrowhead to be at the center point between the sensors that detected it.

In each of our examples from Figure 2.11 we can observe that the traces show the devices moving almost chaotically. They seem to always be moving back and forth. For Assen 2016 we managed to gather comparison data using a GPS. The green line represents the movement of the device as it is recorded by GPS. We can easily observe that the general path of movement is correct, however the trace is much more erratic compared to the real path. The same is likely true for the other three examples.

Although we handpicked the examples in Figure 2.11 to show the circular-movement anomaly, these examples are not exceptions. When movements are traced based on the WiFi remote-positioning data, most of the time, they show this erratic behavior.

Erratic movements, combined with big gaps between few detections, indicate that the output of WiFi crowd-dynamics-monitoring platforms may be lacking and it may be more difficult than previously expected to process this data, in order to harness its information. It is our goal to explore what causes these issues and what is the potential of this type of data.

2.7 Summary

From a multitude of positioning technologies, we have determined WiFi remote positioning to be the one that best techniques matching the requirements of a crowd-dynamics monitoring application. WiFi remote positioning has an accuracy of around 100m and can be used to monitor many individuals simultaneously.

We have built our own implementation of a crowd-dynamics monitoring platform using WiFi remote-positioning and in the process, we have discovered implementation details that have not been reported in the existing literature. The most notable details are the choice of frequency (channel) on which to listen for WiFi frames and the variants of these frames. The type of WiFi frames is particularly important because only some contain the necessary data that lets us identify a device. Furthermore, some frame types, such as Probe Request

frames, have use cases that require regular transmission patterns making them ideal for gathering the multiple positions required to build a trace.

We made use of our crowd-dynamics monitoring platform to conduct five data-gathering experiments. These experiments have provided us with data sets consisting of multiple days of positioning data for crowds of tens of thousands of individuals. These data sets represent different towns, different time periods and different contexts. The rest of our work focuses on analyzing these data sets in order to discover to what extent they can be used to model the dynamics of the crowds that they represent.

Although research literature contains multiple examples of using WiFi remote positioning with promising results for specific applications, analysis of the raw positioning data gathered during our experiments raises some red flags. We have not gathered the same amount of data for all devices, meaning for some we can only detect their presence once while most have few detections, and few have many. Traces drawn using the raw data have large time gaps and exhibit significant anomalies in the form of erratic movements. These issues warn us about possible shortcomings when using WiFi remote positioning for crowd-dynamics analysis.

CHAPTER 3

Understanding difficulties in WiFi-based crowd sensing

By tracing the positioning data we obtain from WiFi remote-positioning systems we can easily notice problems. For some devices, we have few recordings, for others we observe large gaps. When we do have sufficient positions, we observe an abundance of anomalies we call circular movement.

We aim to explore the properties of WiFi remote-positioning systems and determine what are the factors that cause the sparse detections and the anomalies. To do this we analyze the data sets we gathered during our experiments. Considering there is no way to control the target devices and add more data we cannot directly address the sparsity issue, instead we focus on smoothing the traces in order to remove the anomalies.

3.1 Contributions

Because of significant problems like moving in circles, large gaps and generally few detections, visualizations of traces captured through WiFi remote-positioning become fuddled. In this chapter we explore the underlying causes for these anomalies.

In order to truly understand the problems identified through trace visualizations we go into the details of the WiFi remote-positioning systems. **We categorize the properties of WiFi remote-positioning systems and show their effects, as well as discuss the issues we observed**.

Improving the frequency of detections cannot be done in a non-intrusive way. Instead, we concentrate on improving periods for which there are enough detections. Most importantly we address the circular-movement anomalies. To achieve this, **we developed three smoothing algorithms based on RSSI, time**

compression and cycle removal, respectively.

In order to understand the effect and the differences between the three smoothing algorithms, we need to have a way to compare them. **We define two metrics to serve as means to compare the three algorithms: entropy and dissimilarity**. Finally, we showcase the effects of the three algorithms on the trace visualizations.

3.2 Properties of WiFi remote-positioning data sets

A perfect crowd-dynamics model would have strict requirements. It would have to represent the entire population, in order to have no bias. Furthermore, it would require representative movement information. To obtain accurate, representative movement information we require positioning data which has an accuracy significantly higher than the distance of the shortest movement that needs to be represented. Furthermore, the frequency needs to be higher than the duration of the shortest move that needs to be represented. Ideally, this data would be continuous, and not contain gaps in which the position of the person is not known. The shortest movement that needs to be represented is dictated by the application. For example, extracting information on holiday preferences should not require a frequency higher than one detection per day with a positional accuracy on the scale of cities.

Positioning solutions are far from perfect. In the case of WiFi remote positioning we know that positions cannot be recorded for the entire population, as some people simply do not have WiFi-enabled devices. We have no way of knowing how many individuals can be monitored and if the behavior of individuals that can be monitored differs from those that cannot. Furthermore, the positioning accuracy is smaller than 100m (we discuss later in this chapter why this is the case) and we observed in Chapter 2 that detections are sparse and have a frequency that varies significantly (from seconds to tens of minutes). The frequency can also vary from device to device.

We remind the reader that the choice for WiFi remote positioning is based on the fact that WiFi is already widely deployed, in principle making it possible to build a remote-positioning system that scales to hundreds of thousands, or even millions of people. To achieve this, we assume no control over the hardware or software carried by target individuals. Without control of the hardware or software we cannot implement any technique that depends on the targets cooperation in order to increase positional accuracy, frequency or offer any guarantees on the quality of the resulting data sets.

There are three main factors to consider when analyzing WiFi remote-positioning systems and data. These factors are given by the elements of a detection $\langle s, d, t \rangle$, the sensor s, the device d and the time t. These translate in positioning accuracy, device identifier issues, and frequency of detections. We go into details on each of the three, discuss limitations and possible ways of improving them.

3.2.1 Positional accuracy

We know that the signal from a device can be correctly recorded only if the device is in proximity of a sensor. The proximity distance is dictated by the transmission power and antennas. Most commercial WiFi devices have an advertised transmission range of 100m, given line of sight, meaning no obstacles. It is usually assumed that beyond this range the signal diminishes to a point where frames are received corrupted or are simply indistinguishable from background noise.

The transmission medium used by WiFi is far from ideal. The environment has a significant effect on the signal. Tunnels are known to extend the range while buildings and people limit it. Considering this, **the shape and size of the area in which WiFi frames can be correctly received is irregular and varies in time**.

In one of our data sets we have identified 1,491 occurrences in which a device is detected by five or more sensors at the same time. If we were to take the 100m advertised detection distance as an absolute boundary, this would be impossible as there is no location where more than three 100m discs around the sensors can overlap.

What this means for our crowd-dynamics monitoring data sets is that because of the irregularities of the detection area, instead of having a clear separation of when a device is detected or not based on distance, we have an increase in likelihood of a device being detected when it is closer to the sensor. This likelihood is dependent not only on distance but on the angle and changes in the environment. We know of no way to determine this probability, given the large number of parameters that affect it.

Many assume that WiFi remote-positioning data has a fixed positional accuracy that can be further improved. This has been repeatedly shown to be true for indoor environments [99]. For outdoor environments, this could not be further from the truth. The main factor why indoor environments can offer high accuracy is because indoor spaces are usually significantly smaller than the 100m advertised WiFi range. The small distance puts targets in an area where

frames are likely to be correctly received. Furthermore, walls reflect the signal, increasing the chances it is received by a sensor.

Throughout the literature for WiFi remote-positioning for indoor spaces, positional accuracy is increased through the use of trilateration [100, 101]. Trilateration is a well-known technique of combining multiple simultaneously recorded distance measurements in order to obtain positions with increased accuracy. WiFi communication has no way of directly measuring the distance to a device transmitting a signal. However, the measure of received signal strength indicator (RSSI) has previously been correlated to distance [102].

The received signal strength indicator (RSSI) is a value measured by the receiver. It represents the power of the signal when it is recorded. RSSI can be used as an estimate of the distance between the sender and receiver, and by using trilateration [103] or fingerprinting [104], positioning accuracy can be increased from more than 100m to just a few meters. The RSSI-distance correlation is given by the fact that the signal power decreases the longer it travels.

According to the authors of [102], RSSI is closely correlated to distance, assuming the distances are small, under 10m. As such, solutions for improving positional accuracy based on RSSI work for indoor scenarios, where the distance between the sensor and target device is in the order of meters and where walls reflect the signal, lowering the probability of detection when the sensor is in a different room from the device.

Considering outdoor environments, and large distances (tens of meters), the correlation between RSSI and distance no longer holds. Even worse, at a large distance the low signal strength causes frames to be lost. This means that the simultaneous detections **required** for trilateration are rare. This is what we observed in our data sets, where less than 10% of detections were simultaneous.

Outdoor, the signal strength is lowered by more obstacles, such as other humans, trees, cars or buildings. Furthermore, there are many types of smartphones and wireless devices in use today and the power with which a signal is sent differs from device to device. It differs because of the design of the network module or even because of the manufacturing process (impurities in the metal from the antennas). All of these factors add differences to the RSSI values that cannot be controlled.

We found that the frames received from a fixed device can be recorded with very different RSSI values. This variance is well known, and solutions have been proposed. However, these solutions, as in the case of [105], assume many frames are received and an average RSSI value can be calculated. This is possible when the target device is the one controlled instead of the anchors because WiFi access

points transmit *Beacon* frames with a fixed and high frequency. By comparison mobile devices transmit *ProbeRequests* with a lower frequency, affected by the current battery level.

Considering the outdoor scenario, it is impossible to use previous solutions to correlate RSSI values to distance. Consider our Assen 2015 experiment, where we recorded RSSI values, to use as an estimator for distance. First, most RSSI values were very small (-100 to -70), this is because the sensors are installed on posts and are generally far from the mobile devices. Other works that assume a correlation between RSSI and distance use higher RSSI values [102, 106]. Without any available literature, and without ground truth we could not correlate RSSI values to distance. Manual verifications of the traces revealed many cases where high RSSI values were recorded for detections where we believed the device was far (based on surrounding detections at other sensors). Furthermore, distance estimates are particularly useful when simultaneous detections are recorded and we discovered this to rarely be the case, even though the sensors were positioned so that the detection areas have extensive overlap. Because of negative results we stopped recording RSSI values. We did this for the goal of preserving privacy by recording as little data as possible.

We reiterate that we assume we cannot modify the target device. If we could modify the target device, we could improve positional accuracy by controlling the transmission power in order to set the detection range or ensure that every signal is detected by multiple sensors so that we can perform trilateration.

Given the limitations introduced by the outdoor environment, the large distances, the high packet loss, as well as the fact that in most cases we cannot improve the accuracy due to the lack of simultaneous detections, we conclude that WiFi remote-positioning accuracy for the scenarios we consider is limited to that provided by single sensor detections and has a value of around 100m.

3.2.2 Target identifier

It was expected that MAC addresses uniquely identify a device. Unfortunately, this turns out not to be always true. Even if the MAC address is set to be unique when the network module is manufactured, it is common for software to support the modification of MAC address. In practice two devices can have the same MAC address, or one device can have multiple.

Due to privacy concerns some operating systems started randomly changing the MAC address while scanning for networks. The process of repeatedly changing the MAC address to a random one is called MAC address randomization. It is meant as a privacy measure. By obscuring the MAC address it is assumed that

the device cannot be tracked. This technique is used by modern Apple devices [1] as well as Android based ones. However, MAC address randomization can be defeated as showed in [97], where other information, such as sequence numbers, present in the frames is used as unique identifiers for devices. Another technique presented in that paper is the creation of false WiFi access points that force the device to reveal the real MAC address.

In our experiments we make no attempt to find the MAC address of devices that use MAC address randomization and we do not attempt to match multiple such addresses to a single device. We do this to respect the privacy assumed by the owners.

We observed random MAC addresses in all our data sets and the same have been reported by previous WiFi remote positioning experiments [82]. In our data sets we observed the percentage of detections having random MAC addresses grow from year to year, starting at 10% and reaching 35%.

We developed a method to estimate if a device identifier is based on a random MAC address which uses the number of detections per device. The details of the method are presented in Chapter 6. Random MAC addresses are changed frequently and there are many available. Because of this we record a random MAC address only few times.

It is possible that a final WiFi remote-positioning data set contains detections of different devices recorded with the same identifier or having one device be recorded with different identifiers. This can also be true for our data sets. However, we expect this problem to be small especially after we remove the detections for device identifiers that we estimated to be based on random MAC addresses.

3.2.3 Frequency of detections

We found that detections in our data sets are sparse, large gaps are common and their frequency varies considerably. It is common to find detections with a high frequency (every few seconds), for a short duration, surrounded by large gaps or combined with frequencies in the order of minutes.

The main issue is that target devices set their own transmission frequencies. The rate at which frames are sent is based on the battery level, running applications or even the screen status. Furthermore, the device or the WiFi module can be turned off by the owner. Without a fixed frequency or control over the interest device (in order to set a fixed frequency), we have no way of

[1] User Privacy on iOS and OS X - in Session 715 of Core OS WWDC14

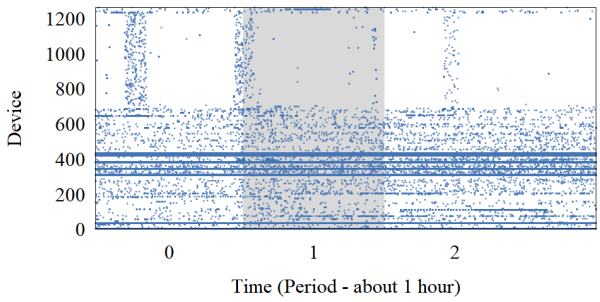
determining if we are not receiving frames because the device is not sending them, the device is not present in the sensing area or because of interference.

During our two-sensor experiment from Chapter 2 we discovered what is likely to be the issue for low detection frequency. The experiment consisted of recording detections with two sensors placed inside a lab room and located only 50cm apart. The room was inside a building with moderate foot traffic in which WiFi seemed to work perfectly (no one had connection issues and there were no issues with bandwidth) and found that many frames are lost due to interference. We observed that about 40% of frames have the retransmission bit set to 1, meaning the receiving device did not confirm the original frame was received and the frame was sent again. This shows that a lot of frames are lost due to interference.

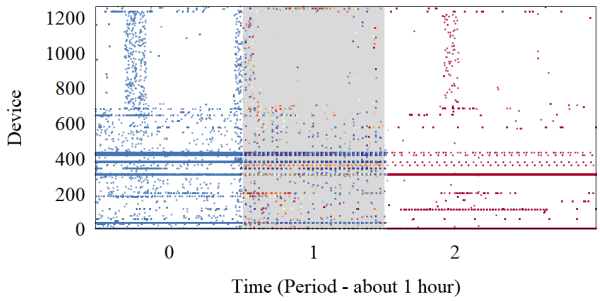
The Assen data sets are built by recording only Probe Request frames. These are management frames which are not retransmitted, making this problem worse. Furthermore, the conditions during our data gathering experiments, large crowds, big distances from sensors, make interference much more likely. The human body diminishes WiFi signals and multiple WiFi devices in the same place increase the chance of simultaneous transmission. When two or more frames are broadcast simultaneously on the same channel they cannot be received. This is called a collision. Frames cannot be received even if only part of them overlaps.

Even worse, by analyzing the data gathered during our two-sensor experiment we further confirm that frame loss is the norm, rather than the exception. We previously analyzed data only from devices for which we had direct control. For the devices inside the room detections were recorded at the same time by both sensors. However, when we consider all devices, devices which are not in the same room as the sensors and because of this have a low signal quality, the results are very different. This can be observed in Figure 3.1. Here, the x-axis represents time, while the y-axis represents the device, each dot represents a detection of the device at the given time. The color of the dot represents the channel on which the detection was recorded. As can be seen in Figure 3.1c, the two sensors gather a significant number of detections for which there is no equivalent captured by the other sensor.

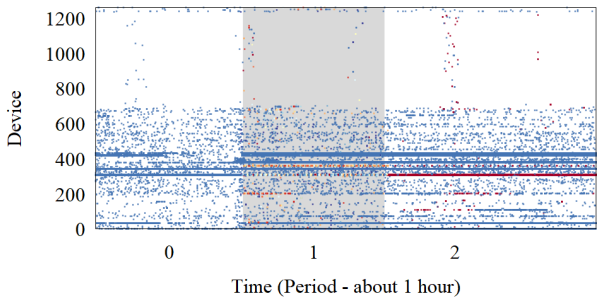
In the figure we present only detections for devices that were seen at least once by both sensors. Other than these, there are a significant number of MAC addresses recorded by only one of the sensors and not the other. Sensor 1 recorded 13479 distinct MAC addresses and Sensor 2 recorded 8244 distinct MAC addresses. Out of these only the 1240 presented in the figure have at least one detection at each sensor.



(a) Sensor 1 (Channel 3)



(b) Sensor 2 (Channel 3 - Channel Hopping - Channel 8)



(c) $\cup \setminus \cap$ of Detections at Sensors 1 and 2

Figure 3.1: Detections from common devices (Each channel has a different color) Two sensors experiment

The MAC addresses between 700 and 1100 all start with the same sequence "da:a1:19". We have identified this to be Company ID (CID), belonging to Google. CID values are similar to OUI values, but they cannot be used to create unique MAC addresses. This means that the value we identified is most likely used by the Android operating system for MAC address randomization. This means that the 400 addresses represent only a handful of devices. The horizontal continuous lines belong to Cisco WiFi hotspots that are offering internet access to the building and the "00:00:00:00:00:00" MAC address, which is probably used by multiple devices.

To better understand the effect of distance on the chance of receiving the frame and recording a detection, we compare subsets of detections from the two sensors. The comparison is shown in Figure 3.2. With green we represent the percentage of detections that have an equivalent in the other data set, and with orange, those that do not. The upper part of the graph represents the detections for Sensor 2 and the lower part, those for Sensor 1. We can extract subgroups from the detections. One group is the set of common devices (devices detected at least once by each sensor), and a smaller subgroup represents the nine devices we had in the lab during the experiment. These would be the ones that have a short distance and are mostly detected the same by both sensors. We use the labels "All", "Common" and "Our" to represent all detections and the two subgroups. Because most of our experiments and many other record only Probe Request frames we extracted this set of detections separately in order to see how it compares to the set of all detections. The same subgroups were extracted.

It can be observed that for the devices in the laboratory, both sensors gather almost the same set of detections. However, when we look at detections from all devices, Sensor 1 has 40% of detections matching a detection recorded at Sensor 2 and Sensor 2 has 80% of detections matching one recorded at Sensor 1. This means that the sensors, which were placed only 50cm away from each other, have very different views. When considering only Probe Request frames, the detections from the two devices are more similar. This is because Probe Request frames are sent in bursts, increasing the chance that at least one is received.

The result of this experiment supports the fact that few WiFi frames are received. This means that **frame loss** explains, at least partially, the low detection frequency.

The high loss rate for WiFi is confirmed by other research [107] that tries to find better estimates for the packet loss rate given different physical layer configurations. Similarly, in their work [108], the authors state that over half of the transmission time for WiFi is used to correct errors. Even so, the authors argue that frame error rates vary. The authors discovered that frames transmitted at

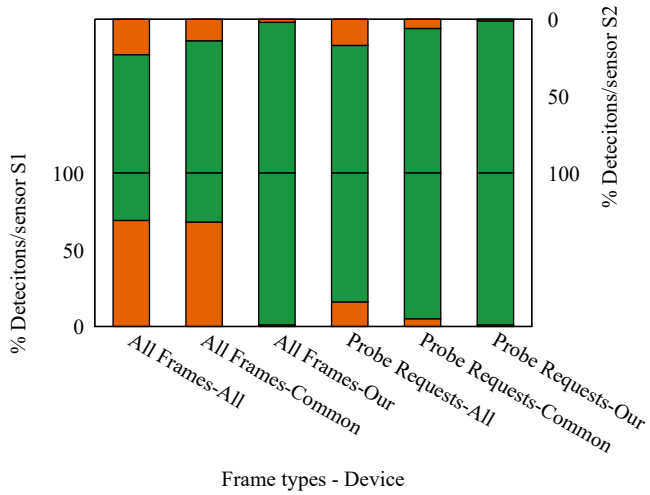


Figure 3.2: Differences between the sensors (green - matching; orange - not matching)

low data rates are more likely to be lost. Considering management frames (such as Probe Requests) are sent at low data rates, so that old devices can still detect them, this raises serious concerns for many WiFi remote-positioning systems.

The authors of [109] show that the high scan rate used by phones becomes a problem when large crowds are formed. The low data rate for transmitting management frames and the large number of devices transmitting in the same area causes throughput issues. These cause collisions and an increase in the number of lost frames. The same results are supported by [110].

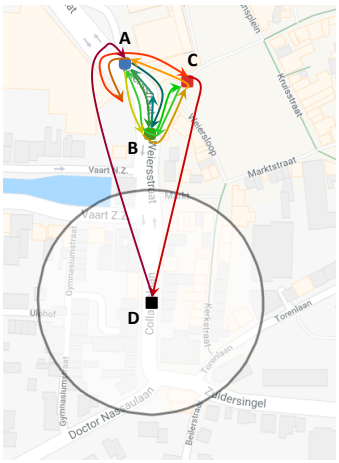
Other than high frame loss, there are cases when detections are not recorded. These cases are far less frequent and less problematic. One example would be faulty sensors. In ideal circumstances sensors would never break and would continuously record data. Sensors can fail, permanently or temporary. In our experiments, such a gap is present because the sensors are configured to perform an automatic reboot every 24 hour. Other off-line times are also present, although rare.

Although not affecting frequency, timing errors can create the circular-movement anomalies. These errors are very rare. When a detection is recorded the sensor records the time stamp for it. The time stamp is based on the internal clock of the sensor. For the data set to be correct, the internal clocks of sensors

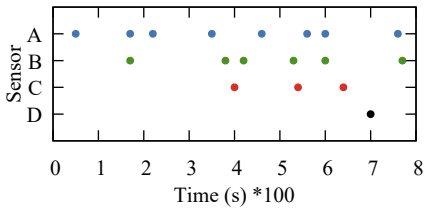
need to be synchronized, otherwise it would be impossible to determine a global ordering of detections. As previously mentioned, we make use of the Network Time Protocol to synchronize the clocks. Even with synchronized clocks it is possible, although very unlikely, to have irregularities. Two different sensors that receive a frame near the transition between two seconds may record it with different time stamps. When tracing the path this could make the device seem to be moving in the wrong direction.

3.2.4 Explaining the anomalies

Consider a static device, its detections are presented in Figure 3.3b. If we trace the detections of this device we can draw the path from Figure 3.3a. As a visual guide, the path starts with dark green and changes to dark red towards the end. This example illustrates what we found to be common in our data sets. Tracing based on the WiFi remote-positioning data reveals circular-movement anomalies, even for devices that are static.



(a) Path (circle is 100m visual guide)



(b) Detections Over time

Figure 3.3: Irregular movements of a static device (artificial trace based on real ones)

We believe our example device is placed between sensors A, B and C. In ideal circumstances every frame sent by the device would be detected by all three sensors. This is rarely the case. The device is detected by only one sensor at a

time, at most two, and the order of detections seems to be random. Furthermore, occasionally, the device is detected by a fourth sensor that is placed further away. A first look at the trace gives the impression that the device is constantly moving randomly, between the three sensors, and sometimes to the fourth.

We identified multiple such examples in our data set, where a device that is static appears to be moving similar to what we present in Figure 3.3. We confirmed that these devices are static because the OUIs match those of WiFi routers or printers and considering that these devices are constantly detected by the same two or three sensors for large periods of time (most of the day).

Most traces we extract from our crowd-dynamics monitoring data sets have these problematic characteristics. The problems get even worse for mobile devices. Instead of a clear path between the start and the destination, we see the device moving back and forward having only a general trend of approaching the destination. This movement seems erratic.

The main reasons for this observed behavior are a high number of *lost frames* and the low positional accuracy given by *dynamic, irregular detection range*. The positioning accuracy is irregular because the signal range is extended by streets and limited by buildings and even people. The frequency is primarily affected by the high number of lost frames, due to interference and low signal quality. Low signal quality is caused by the large distance between the target device and the sensor as well as buildings and people blocking the line of sight path. Interference is caused by the large number of people, and with them, devices that transmit at the same time using low bandwidth (and as such occupying much of the transmission time).

This characteristic is not particular to our data sets. Similar works [80] exhibit the same characteristic in their visualization. This is also the case for GPS positioning [111], although the problem is not as severe.

In the case of GPS data sets the circular-movement issue has been addressed through smoothing [112]. The example trace presented by the authors displays a back and forth movement similar to the one we encounter. The smoothing is done by applying outlier removal (where unlikely positions are removed), interpolations (by adding new detections). Similarly, the authors of [113] smoothen the traced GPS path using outlier removal.

Outlier removal cannot be used to smoothen traces obtained through WiFi remote positioning. Outlier removal works for GPS because one position can be erroneously recorded to be far from the others in the trace. For WiFi, the set of possible positions is small and erroneously recorded positions are around the area where a device is located.

3.3 Smoothing traces

Information retrieval from mobility data sets that have small positional accuracy and low detection frequency, is not trivial. Even simple tasks such as differentiating between mobile and static devices requires expert intervention.

Visualizations of individual movements are unclear. By tracing the detections of one device we see it randomly moving back and forth between sensors. Only careful analysis, through tedious visual-temporal inspection, reveals a general path.

Most properties described in the previous section can be tackled by having control over both the target device and anchors. We want to preserve the assumption that there can be no control over the target device. Yet, to be able to extract useful information we need to address scenarios such as the one presented in Section 3.2.4. We do this by modifying the raw traces to smoothen the paths and remove irregular back and forth movements.

Smoothing the traced paths improves the visualization system and enables complex information retrieval. Take one simple question "How many times did people move from sensor A to B?". If many traces contain back and forth movement between these two sensors, the answer to the question would be skewed. One movement appears as multiple.

We define three methods that identify a subset of detections, which when removed or modified, simplify, or in other words, smoothen a trace. The goal is to obtain smooth traced paths that retain the general shape of the original ones.

3.3.1 Detections with low RSSI values

When the transmitting device is far from the sensor, or the environment is very noisy, frames are received with a small power level. This is measured by the RSSI value.

The intuition is that detections with small RSSI values are of low quality and that the trace would be improved by removing or modifying these detections.

There is no known threshold under which we can say a detection is of low quality and should be removed/modified. We note this *threshold* with R and in order to find the appropriate one we test multiple values for it.

Removing detections based on RSSI has been done before [114]. However, the authors offer no discussion on how the threshold is chosen. We believe there is no one-size-fits-all solution because the RSSI values are affected by multiple factors including transient ones, like weather.

3.3.2 Frequent detections

The speed of pedestrians does not vary much from one person to the other. We can make use of this range in selecting useful data. By manually analyzing traces from our data sets we concluded that people change their position rarely and as pedestrians, slowly. We can say that the frequency of significant location change is lower than the detection frequency. Because of this we assume that there exists a size for a time interval so that each time interval has one detection that gives the position of the device for the entire interval.

To determine these dominant detections, we split the time into intervals of ΔT seconds and select one dominant sensor for each interval. All detections belonging to other sensors are removed or modified to contain the dominant sensor. We say a sensor is dominant if it has the highest strength according to equation 3.1. The strength represents the sum of RSSI values in all detections inside the interval belonging to the sensor-device pair. We choose the sum of RSSI values because we expect a close to linear correlation between the RSSI value and the frame loss rate. This means that many detections with low RSSI values are stronger, more representative, than one detection with a high RSSI value.

$$Strength(\mathsf{S}, \mathsf{d}, \Delta T) = \sum_{\langle S,d,t\rangle, t \in \Delta T} |RSSI| \tag{3.1}$$

Here, $RSSI$ represents the signal strength of detection $\langle \mathsf{S}, \mathsf{d}, t \rangle$.

A similar mechanism to simplify traces is used by the authors of [80]. In their work, the authors sample the trace by keeping one detection every ΔT seconds. They do not use any mechanism to ensure that the chosen detections are representative. Similar to the RSSI case, the authors do not offer a discussion on how to choose the value for ΔT.

Although these techniques are also found in other works from the literature, as to our knowledge, we are the first to present a comparison between the usage of multiple values for ΔT or R.

3.3.3 Cycles in the path

As previously stated, it is common for traces to display back and forth movements in a sort of circular behavior, caused by lost frames, low transmission frequency and irregularities of the transmission range. We propose a direct way to combat back and forth movements. We identify a back and forth movement in a trace as a set of consecutive detections that has the first and last detection at

the same sensor and no more than X detections at other sensors before the first sensor is detected again.

We must set X so that we remove the back and forth cycles yet maintain as much information from the original trace. For instance, a value of X that is too large would identify natural cycles in movements, such as going to a shop and back, as irregular ones. In contrast, a small value of X would identify no cycles. We test our solution with multiple values for X.

For each cycle we mark for removal or modification all detections recorded at sensors that are not dominant. The dominant sensor is identified using equation 3.1.

It is possible for a detection to appear in multiple cycles. In this case the cycle with the earliest start time is chosen. By choosing the cycle with the earliest start time we give an advantage to the cycle with the longest history. Algorithm 1 shows the steps required to identify the back and forth circular movements using this method.

```
Result: marked detections
detections;
for each device do
    identify cycles
    for each cycle do
        find sensor with highest Strength
    end
    for each detection do
        get earliest cycle containing detection
        if sensor != cycle dominant sensor then
            mark detection
        end
    end
end
```

Algorithm 1: Cycle identification

3.4 Comparing trace-smoothing techniques

Measuring the correctness of a trace is done by comparing it to the movement of the individual that generated it. The movement can be recorded using more accurate systems, such as GPS or by keeping logs. We consider the recorded,

accurate movement, to be the ground truth. Collecting ground truth on the scale of our experiments is not feasible. Because of this we need to find other techniques to verify our results.

One way to compare the efficiency of the rules is to trace the paths taken by individuals, before and after we modify them, and conduct visual comparisons. However, although simple, this method can be subjective, and it requires manually going through many traces. Furthermore, traces can be long and complex, requiring the analysis of animations instead of images. We conduct this type of verification on a small sample of traces to confirm our results.

As an alternative to visually verifying a modified trace we propose two metrics. These metrics can be applied on the entire data set and they offer different insights on the results. One is entropy and measures how much our techniques smoothens the trace. The other, we call dissimilarity and it measures how different the modified trace is from the original. We use these metrics because the goal of our techniques is to remove irregularities while preserving the general path.

To calculate the **entropy** we take all pairs of two consecutive detections of a device and we calculate the probability that the second detection occurs at a specific sensor, given the sensor at the first detection. We model it as the Shannon entropy [115], defined in equation 3.2.

$$H(\mathsf{S}) = -\sum_{S^*} p(\mathsf{S}^*|\mathsf{S}) \log\ p(\mathsf{S}^*|\mathsf{S}) \tag{3.2}$$

Here, p represents the probability that a device triggers a detection at sensor S^* as the next one after a detection at sensor S. The entropy is calculated for each sensor S and averaged.

Dissimilarity measures the difference between the smoothened trace and the original. Given two traces, with the same number of points and with matching time stamps, we define dissimilarity to be the average Euclidean distance between the positions of sensors at matching detection times in the two traces. To offer a concrete example, if we had two traces, with a single detection for each, with the same time stamp, one triggered at sensor A while the other triggered at sensor B, the dissimilarity value would be the Euclidean distance between the two sensors. Given a trace, it would be the average distance of all such pairs.

A large dissimilarity value represents a large disturbance in the path. This means that if we make big changes to the original path, the dissimilarity value would signal this. In contrast, if the dissimilarity value is small it means the difference between the original path and the modified one is small.

The goal is to have a small dissimilarity value, so that the simplified path does not lose information contained in the original one. But dissimilarity by itself is not sufficient. The smallest dissimilarity is zero, meaning the traces are identical. We want to simplify the path, so dissimilarity needs to be used along with the entropy metric.

The three methods we described earlier, based on RSSI, time and cycles identify detections that have low quality. We can smoothen the data set by dealing with the low-quality detections in two ways:

- **Remove** low-quality detections.
- **Modify** the detection by assigning it to the closest dominant sensor. The closest dominant sensor is determined differently based on the smoothening method.

We calculate the **dissimilarity** on the data sets for which we **modify the sensors**. We do this because the dissimilarity function requires the same number of detections.

We calculate the **entropy** on a data set obtained by **removing** low-quality detections. This gives us a more realistic entropy value. When we modify detections to have the dominant sensor, we create many consecutive detections with the same sensor, lowering the value of entropy.

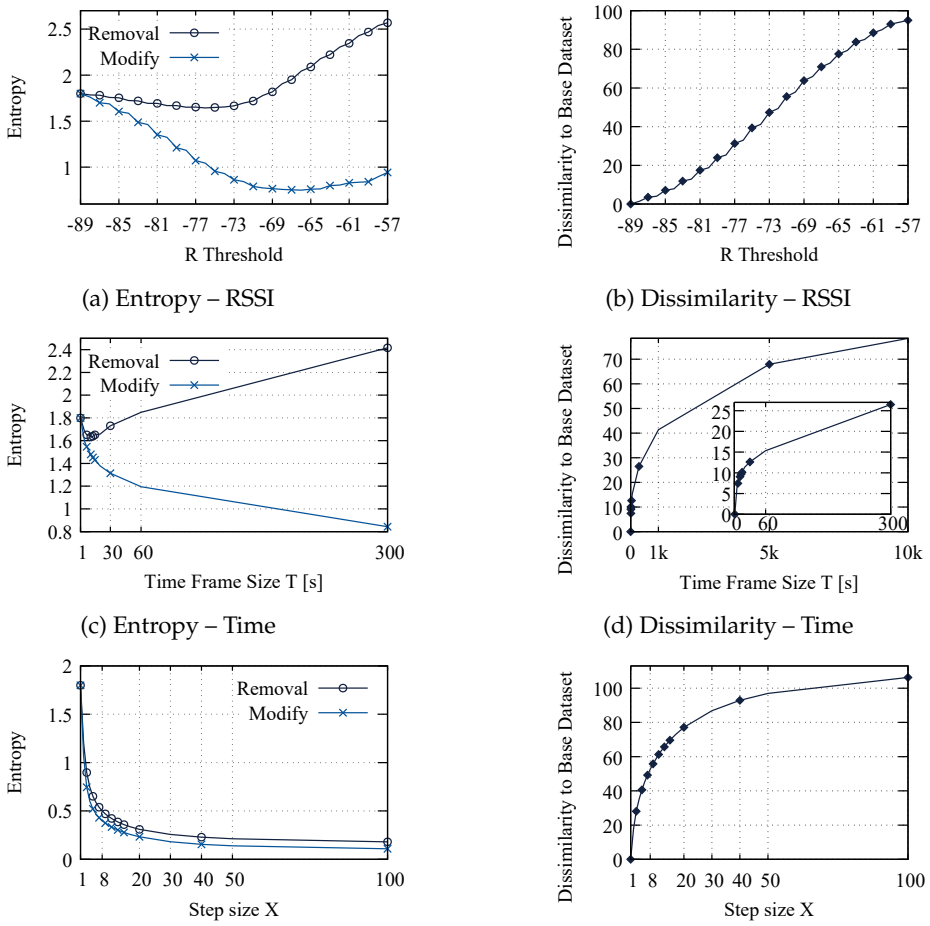
We compare the trace-smoothing techniques using the *Assen 2015 data set*. Each technique has one variable that controls the aggressiveness of the smoothing, R for the RSSI method, ΔT for the time method and X for the cycle method. To conduct a thorough comparison, we iterate through appropriate values for these variables.

The ideal result would have small values for both entropy and dissimilarity. This means that the new trace is less random (no going back and forth), while preserving the general shape of the original.

The RSSI values in our data set are between -21 (strong) and -89 (weak), with more than 95% of detections between -57 and -89. We set R to take all values between -57 and -89. This ensures that we test all relevant thresholds.

ΔT can take values from 1 to infinity. We test with small values as well as a span of larger ones. We concentrate on small time frames as these will show the most variation. If the time frame is too large, the changes will hide most of the fine movement we are interested in.

X can take any positive integer as a value. We found it works best with small values. We set multiple small values but go all the way to 100. Again, if we use



(a) Entropy – RSSI

(b) Dissimilarity – RSSI

(c) Entropy – Time

(d) Dissimilarity – Time

(e) Entropy – Cycle

(f) Dissimilarity – Cycle

Figure 3.4: Entropy and dissimilarity values after using the RSSI, time, and cycle smoothing techniques

large values, the smoothing method will hide most movements. We found 100 to be sufficiently large to show the trend of setting this threshold.

3.4.1 Entropy results

To compare the smoothing algorithms, we make use of the Assen 2015 data set. We apply each of the smoothing techniques, using all the thresholds presented above for all the traces in the data set. We do not perform any outliers-removal or any other modifications to the data set and the traces. Figures 3.4a, 3.4c and 3.4e show the calculated value of the entropy based on each setting of the variable that controls the aggressiveness of the technique in terms of how many detections are selected to be modified or removed. With an increase in aggressiveness we observe a decrease in the entropy of the resulting traces. This is expected. As the traces become smooth, there is less randomness in the order the sensors appear in the trace.

If we keep increasing the X value for the cycle method the entropy continues to go down, until all cycles are removed from all traces, be it real or anomalous. This is not the case for the RSSI or time methods. When we increase the aggressiveness for these methods, we reach a point where the entropy starts to increase. After this point, the selected detections for removal or modification are representative of the general path. When we remove or modify them, we create gaps in which a device jumps from one sensor to one that is far away.

3.4.2 Dissimilarity results

It is not enough to smoothen a trace to a point in which entropy is low. When we lower the entropy, we risk losing information by missing steps in the general path. We use dissimilarity to measure how much we modified the traces.

Figures 3.4b, 3.4d and 3.4f show how the dissimilarity between the original traces and the smoothed ones increases with an increase in aggressiveness for each of our methods. This is true regardless of the method used to smoothen traces.

3.4.3 Comparing the results

In order to better compare the techniques, we choose what we believe to be the best aggressiveness setting for each one and compare the entropy and dissimilarity values for these settings. For the RSSI method we set the R threshold to -75, representing the point of lowest entropy. We apply the same thinking to the time-based method, and we choose ΔT to be 11 seconds. For the cycle-based method entropy continues to go down with an increase of aggressiveness while the dissimilarity continues to increase. This means that the point of lowest

entropy is also the one with highest dissimilarity. We choose X to be 4 as the point where the rate of drop in entropy becomes almost constant.

We choose the threshold to be one where the entropy is lowest because this implies that the maximum amount of noise (cyclic movement) has been removed from the traces. When the entropy increases again for the first two smoothing methods, this is because detections that help in defining the path are removed and consecutive detections appear at far-away sensors.

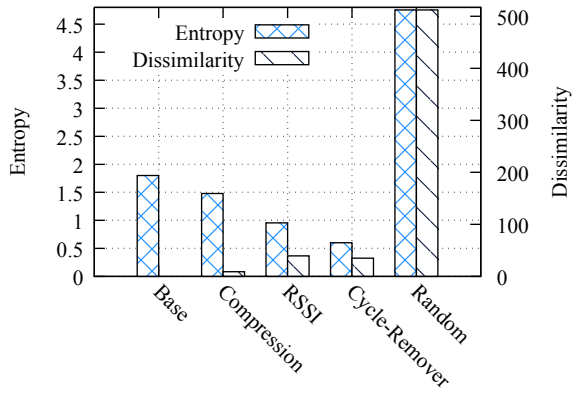
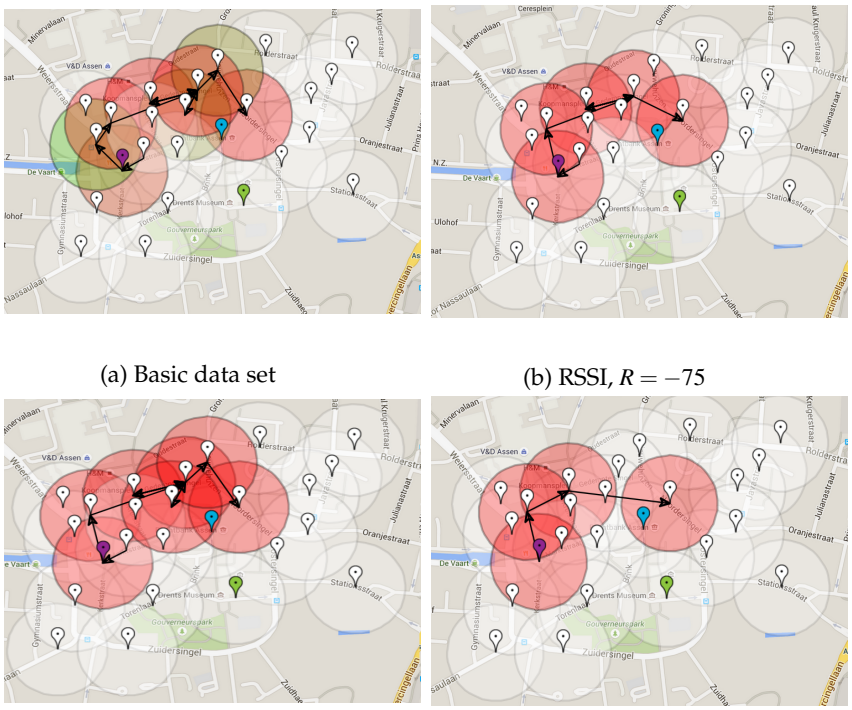


Figure 3.5: Comparison between smoothening techniques, as well as the base and random generated data set

The entropy and dissimilarity for these aggressiveness values is presented in Figures 3.5. For a better comparison we add the base data set and a randomly generated one. For the base data set we can see the entropy value, highest compared to the smoothed traces. We calculate the dissimilarity by comparing it to itself, resulting in the lowest value, zero. The randomly generated data set has both the highest dissimilarity, compared to the base one, and the highest entropy.

We extract one device from the data set and trace its path before and after smoothing using each of the three techniques. The aggressiveness values are set to the ones presented above. The path for a 10-minute window can be observed in Figures 3.6. The green and red circles represent sensors that detected it. The green ones represent detections with low RSSI values. The arrow shows the ordering of detections. An arrow can end between sensors when multiple sensors detect the device at the same time. In this case we make the arrow end



(a) Basic data set

(b) RSSI, $R = -75$

(c) Time, $\Delta T = 11$

(d) Cycle, $X = 4$

Figure 3.6: 10-minute path made by a device (the circles are 100m visual guides)

in the geometrical center of the positions of the sensors that detected it.

In the base data set we can observe the back and forth behavior we previously mentioned. Even though the RSSI and time-smoothened paths contain less detections, and they both remove the low-quality ones based on RSSI, the back and forth movement is still present. This is not the case for the cycle solution. In that case, even though the path contains fewer detections, all the remaining detections are representative of the general path.

This example was specifically chosen from our data set. The results are not the same for all traces, but we have found that most are similar. The cycle method we proposed appears to be the most efficient in removing noise from the traced paths.

Another argument in favor of the cycle-detection method is that it has the highest increase in the number of static devices after the traces have been modified to have the cycles removed. These devices had traces that contained detections at multiple sensors. After the removal of the marked detections, the trace contained detections at only one sensor. We manually verified a few traces for devices which were identified as static after applying this method and we discovered that the traces had detections similar to the ones in Section 3.2.4, appearing to be devices that have constant, circular movement.

3.5 Summary

Crowd-dynamics monitoring platforms require data from a significant part of the population and in order to gather data from many individuals they need to be unintrusive. WiFi remote positioning is unintrusive because WiFi-enabled devices, such as the smartphones that we always have with us, broadcast information revealing our position.

This requirement of remaining unintrusive introduces some considerable limitations to our crowd-dynamics monitoring platform. Systems that have control of the target device can implement techniques that alleviate or solve the following issues:

- **The frequency of recorded positions is generally low, varies significantly and includes large gaps.** Positions are recorded at the frequency at which WiFi frames are captured. But the frequency with which these frames are sent is controlled by the target device. These frequencies depend on the applications installed on the device as well as the usage patterns of the user. Different devices can have different frequencies and energy storage limitations have pushed operating system developers to lower the frequency as much as possible. To make matters worse, there is no way to know if a position is not recorded because no frame has been sent, because the target device is out of range or because of interference and we have shown that interference makes us lose a significant portion of frames transmitted in the sensor range.
- **The positioning accuracy is low.** In some cases, we have discovered that frames can be recorded at distances higher than 100m, although this is not generally the case. Furthermore, the positioning accuracy cannot be improved through trilateration because of the low percentage of simultaneous detections of a device at two or multiple sensors. With a

positioning accuracy of around 100m many short movements cannot be identified when trying to trace them using the WiFi remote-positioning data.

- **Individuals may be represented in the data set with multiple device identifiers.** In order to preserve the privacy of their users, modern smartphone operating systems randomize the MAC addresses used during the scanning process.

These limitations are the cause of circular-movement behavior which we observed in the traces drawn from the WiFi remote-positioning data. Because of these anomalies, it is difficult to perform simple tasks such as differentiate between mobile and static devices.

We propose three different techniques to smoothen the traces and remove these anomalies. We have validated our results using a measure of entropy and one of dissimilarity as well as manual inspection of a sample of traces before and after the smoothening algorithms have been applied.

CHAPTER 4

Identifying movements

Positioning data can be used for individual analysis, but when we try to model complex crowd information, such as flows, the use of raw data becomes an obstacle. Throughout the tracing and positioning literature we identify a common abstraction: superfluous traces can be summarized into periods of stops and moves.

Periods of stops and moves describe individual dynamics in a concise manner. This enables simple and efficient information extraction. Questions such as: "*How many people visited place B right after place A?*"; "*How does the flow starting at location A split after it reaches location B?*"; "*How do the flows differ between morning and afternoon?*"; and many others can be simply modeled as queries on sets of stops and move periods.

Periods of stops and moves cannot be naively extracted. Detecting a device at two different positions does not mean it moved as this can be caused by the low accuracy of the positioning technique. This is evident for GPS, where a device placed inside a building records positions around that building, appearing as very mobile over short distances. This movement is equivalent to the circular-movement anomalies.

4.1 Contributions

We have identified three algorithms that extract periods of stops and moves from GPS traces. We selected these algorithms so that they use different characteristics of the trace: speed, direction, distance.

In order to **validate and measure the performance of these algorithms on WiFi remote-positioning data we conducted a small data-gathering experiment for which we collected ground truth**. This experiment was conducted as part of the Assen 2016 data collection.

We show that we can extract predominant characteristics from WiFi remote-positioning data which we can then use **to bring improvements to the algorithm based on distance**. The algorithm uses geographical distance which does not consider the map of the city or the paths people prefer. Our solution replaces the distance function to better reflect what can be observed from the data.

Instead of geographical distance **we propose three metrics that can be used to model the "closeness" of sensors**. These metrics are based on the positioning data itself. Furthermore, because they replace the distance function their use enables the algorithm to be executed on traces for which the location of sensors is not known.

4.2 Detecting Movements

There has been significant research conducted on extracting information out of GPS traces. In contrast, WiFi remote positioning is a recent technology and the research on information extraction from the generated data sets is still limited. We searched the literature on information retrieval from GPS traces in order to see what elements can be applied to traces collected using WiFi remote positioning.

A vital processing step for GPS traces is the extraction of stop periods. By identifying stop periods, we split the trace into periods when the device is stopped and those when a device is moving. With this information we can identify interesting locations flows and we set the groundwork to answer more complex questions.

This is not a trivial task because GPS does not work indoor and offers several meters accuracy outdoor, when a device stops or enters a building the trace contains positions that are randomly spread in the area around the device. In contrast, when the device is moving the points appear in a jagged line between the last stop location and the next one. The traced line of consecutive positions is jagged because of low positional accuracy and small movement speed of pedestrians.

We selected three algorithms that identify stop periods in GPS traces: **Cb-smot** [116], **Dbsmost** [117] and **Stay Point Detections** [118]. Multiple similar algorithms exist in the literature but these three are the most popular and make use of different characteristics of the trace: distance, speed, and direction of movement.

The algorithms perform well on GPS data sets, but to our knowledge, we are the first to measure their performance on WiFi remote-positioning traces.

The main two differences between GPS and WiFi remote-positioning are: (1) positional accuracy, from 5m for GPS [119], to over 100m on WiFi and (2) detection frequency, which is fixed for GPS but can vary widely for WiFi. As we showed in the previous chapters, positions obtained from WiFi remote-positioning systems are interchangeable with the location of the sensor. Because of this, the only step we are required to take to apply the algorithms on our WiFi remote-positioning data sets is to replace the sensor ids with their positions.

The **Cbsmot** [116] and **Dbsmot** [117] algorithms use clustering to identify points gathered around stop positions. Each cluster represents a stop, while all points not in a cluster belong to a movement period. Both algorithms make modifications to the dbscan clustering algorithm [120].

Dbscan clustering works by taking all the points in the data set and calculating the distance between any two pairs. Usually the Euclidean distance is used. If the distance between two points is smaller than a threshold ε, the two points are considered neighbors. Points with more neighbors than a limit $minPts$ are considered core nodes and are used to expand clusters. All neighbors of core nodes will belong to the same cluster. If any of these neighbors are core nodes, their neighbors will also be added to the same cluster. The process repeats until all nodes are labeled to be part of a cluster or are labeled as noise.

Both cbsmot and dbsmot add restrictions to the original dbscan clustering algorithm. In both cases positions must belong to consecutive detections to be considered as part of the same cluster.

Cbsmot uses both the time difference and the geographical distance between consecutive points as a distance function. By using both time and distance it clusters points at which the device has a low speed. Dbsmot uses the change in direction of movement as a distance function. The thinking is that when points are grouped together the direction of movement changes frequently. This is on par with the circular-movement anomaly.

Stay Point Detection [118, 121] is the simplest of the three algorithms. It starts with a pivot at the first position, iterates through the next points and updates the pivot to a new position when it finds one further than a set distance threshold from the pivot. The thinking is that is we can find a threshold, so that when positions are further than that set threshold, the person must have moved.

4.3 Algorithm Comparison

During the Assen 2016 data-gathering experiment we collected ground truth for a small number of devices. We formed a group of four people, carrying nine WiFi-enabled devices (smartphones and tablets) and walked through the Assen city center on two consecutive evenings during the festival. To collect the ground truth, we made notes of our movements and had two of the devices constantly record their GPS position. We later analyzed the positions and made a list of periods when we were moving and periods when we stopped.

Not all stop periods are relevant. Consider walking towards a shop and having to stop at several red lights on the way. We argue that only the stop at the shop is relevant as that is the goal of our movement. These relevant stops usually have a longer duration. The minimal duration of stops determined by each algorithm depends on their parameter settings as well as the accuracy and the density of detected positions. Because we manually label the list of stop and move periods representing ground truth, we consider only stops that have a duration longer than five minutes. We determined empirically that under this duration it is difficult to correctly label all stops.

We can set different parameters depending on the algorithm. For Cbsmot we can set *Max distance* and *Min time* to control the maximum speed at which the trace indicates the person is "moving" to consider the period a stop. For Dbsmot we set the *Min direction change* and the *Max tol*. Finally, for Stay Point Detection we set the *Min distance*. The values for these parameters are presented in Table 4.1.

Table 4.1: Parameter values

Parameter name	Values	Measuring unit
Max distance	50, 60 ... 490, 500	meters
Min time	60, 120, 300, 600	seconds
Min direction change	15, 30, 45, 60, 75, 90, 120, 150	degrees
Max tol	1, 2, 4, 6, 10, 20	
Min Distance	50, 60 ... 490, 500	meters
Max move duration	1800, 2700, 3600	seconds
Min stop duration	0, 20 ... 300, 600 ... 3300, 3600	seconds

We execute each algorithm on the raw data of the nine traces and select all stop periods that are shorter than a *Min stop duration* threshold and re-label them as move periods. This is done in accordance to how we select stop periods

for the ground-truth. Furthermore, these short stop periods would clutter our results, making it difficult to extract information. We deal with the large gaps by using the *Max stop duration* threshold. If a move described by only two detections has a longer time than the threshold between the two detections, we remove it. If the threshold is large enough it ensures that a stop has taken place during the period that would have otherwise been labeled a move.

After we extract stop and move periods for our nine devices, we compare the results to the ground truth. Because there are only two possibilities, moving or stopped, we consider this to be a binary classification problem. Periods marked as moves are considered positive and all other are negative. For each algorithm, we take each of the nine generated sets and compare each to the ground truth. Because the sets have different time resolutions, we compare on a per second basis. Every second of correctly identified movement is considered to be a true positive (TP), all others are false positive (FP). All seconds of correctly identified stop periods are considered to be true negatives (TN), while all others are false negatives (FN). We then use these values to calculate the F1 score 4.1. We tested each algorithm with multiple parameter settings (for the parameters presented above) and extracted the results for the parameter setting that had the highest average F1 score for each algorithm. The results are presented in Figure 4.1.

$$F1Score = \frac{2 * TP}{2 * TP + FP + FN} \tag{4.1}$$

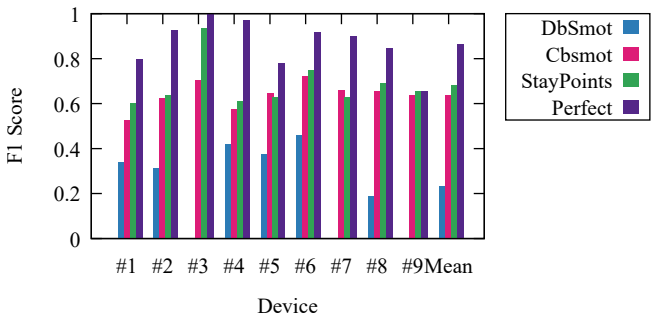


Figure 4.1: Comparing algorithms

Because of the large gaps between consecutive detections, even if our algorithm was perfect, we cannot generate a set of stops and moves identical (with a per second accuracy), to the one from ground truth. To show this, we added

in Figure 4.1 next to the measured F1 scores of the three algorithms the "Perfect" result. "Perfect" is an algorithm that uses the ground truth to label detections as belonging to a stop or a move period. It then groups detections to form a set of stops and moves. Because of the gaps between detections, state changes between two detections are mislabeled. Considering the "Perfect" algorithm does not reach an F1 score of one, it is impossible for any other algorithm to do so. "Perfect" represents the ideal upper limit.

The least accurate method is Dbsmot. Unlike for GPS traces, WiFi has a limited set of available positions. Without trilateration (which requires detections at multiple sensors with stable RSSI values), the number of available positions in WiFi traces is equal to the number of sensors. It is possible to consider positions between two or more sensors even without trilateration, but that still requires simultaneous detections. Simultaneous detections are so rare we decided to dismiss them and simply consider the set of possible positions to be the set of locations of the sensors. In contrast, GPS offers an almost continuous positioning scheme for the entire earth surface. Because of the limited set of positions, there are frequent direction changes. It takes the length of a handful of sensor detections areas, to move from one side of the city center to the other, at which point the direction must change significantly for the movement to continue. When we analyze GPS traces we can observe that due to the high frequency of positions, the direction of movement does not change much, however, when the person has stopped positions are recorded seemingly random around the individual making the trace contain a high number of large angle direction changes.

The other two algorithms, Cbsmot and Stay Point Detection, have similar results. However, Stay Points is much simpler to implement and runs faster. For our data set, running the algorithms with all the parameter settings from Table 4.1, the execution time was in the order of minutes for Stay Point Detection and in the order of days for Cbsmot. Dbsmot is even slower because of the multiple computationally intensive calculations for direction of movement.

4.4 Algorithm Robustness

Algorithms for extracting stops and moves do not offer perfect results. This is especially true for WiFi remote-positioning data, where the low frequency and low positional accuracy makes it difficult even for humans to determine if a person is moving or is stopped based solely on the positioning data. The accuracy of the algorithms is dependent on walking speed and frequency of

detections.

Our aim is to explore the effect that differences in walking speed and frequency of detections have on the accuracy of the three algorithms. Having this information allows for targeted deployments based on the application and context.

We argue that it is impossible to set a real-life experiment in which to precisely control the walking speed of many people. As an alternative, we build a WiFi remote-positioning data set based on simulated movements and detections. This allows us to control the speed as well as the frequency of detections.

4.4.1 Generating a synthetic WiFi remote-positioning data set

We simulated the movements of 100 individuals over the street map of the Assen city center (one of the locations used in most of our data-gathering experiments). We placed sensors in the same positions as they were placed during the 2016 festival and recorded detections whenever an individual was within 100m of a sensor. We use the 100m value as it is the advertised WiFi transmission range. However, there is no other element in our simulation that assumes the use of WiFi. The data could be generated by any repurposed communication protocol remote-positioning technology or any radio-based positioning system. This makes the results and analysis easily transferable to a different technology.

We used a movement model similar to *randomwalk*. The simulation starts by placing everyone at a random location on the street map. For each of the 100 individuals, we select another location which is more than 300m away from the current one and let her follow the shortest path towards it. Once the new location is reached another one is chosen based on the same criteria. We chose 300m as it eliminates a lot of short movements which we know we cannot detect, and it provides a good trade-off between the sensing radius and the scale of the entire sensing area. The individuals walk for one hour and stop for one hour. This is repeated for 24 hours after which the simulation ends.

Individuals move using a fixed walking speed. We split the 100 individuals in groups of 10 and give each group a speed between 0.1m/s and 2m/s. The normal human walking speed is 1.4m/s, by comparison. The results are averaged over each group of 10.

We gather detections every second. This results in a data set which has, on average, 2.5 detections per second per device. The number is bigger than 1 mostly because a device can be simultaneously detected by multiple sensors. To simulate different frequencies, which are affected by the chance of frame loss, we randomly sample the entire data set of recorded positions. As such,

we create 10 data sets by sampling between 0.01% and 100% of detections (the percentages are chosen on a log scale).

To reiterate, we have 10 data sets, sampled based on different percentage of devices, consisting of 10 groups of 10 individuals each, walking at a fixed speed for the group. This results in 100 different settings for speed and detection frequencies.

4.4.2 Results - simulated data

We run the algorithms on the data sets and compare the results with the ground truth. We average the accuracy of correctly labeling all one second periods, over 10 representing the number of devices for each speed setting. We do not expect perfect accuracy. This is because individuals can take paths that lead them outside the sensing area.

Each of the algorithms that we compare has different parameter settings. For each we chose the parameter setting that maximizes the average accuracy. This enables us to conduct a fair comparison.

Figure 4.2 shows the F1 Score accuracy values that we calculated for the Stay Point Detection algorithm. The highest accuracy is achieved when the speed and percentage of detections are high. This is because it is simpler to differentiate between stops and moves if the moves have a high speed, and if we have a lot of data.

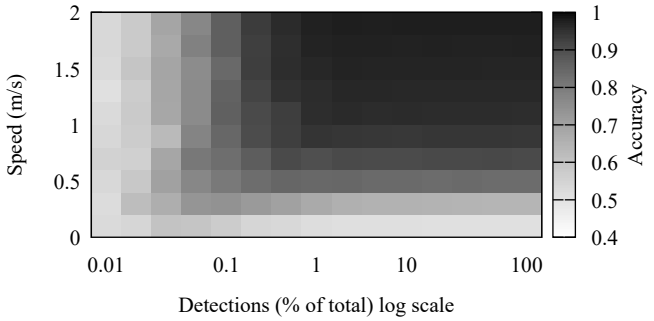


Figure 4.2: Effects of speed and detection frequency on the accuracy of Stay Point Detection

The number of detections has a smaller effect compared to that of speed. We do note that real-life detections are generated at intervals that better match be-

tween 0.1% and 1% of detections recorded every second. Even in the simulated case, the accuracy drops significantly when the number of detections is so low.

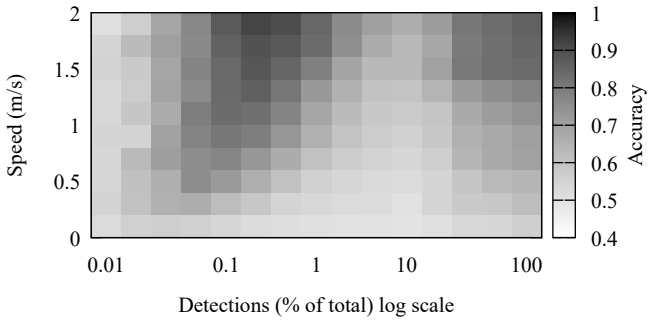


Figure 4.3: Effects of speed and detection frequency on the accuracy of Cbsmot

The accuracy values of the Cbsmot algorithm are presented in Figure 4.3. What is interesting is that this figure presents two different areas with high accuracy values. This is because the algorithm is based on clusters that represent stop periods. When the number of detections is low, smaller clusters are formed because consecutive detections are recorded at far-away sensors. As such, the accuracy of the algorithm for labeling moves is increased. In contrast, for many detections, the accuracy is high because many simultaneous detections force the separation of clusters and labeling of more moves.

We did not apply the same procedure for Dbsmot. The algorithm uses the most computational power and the accuracy for the case of high detection rate and high speed was considerably lower compared to the accuracy of the other two algorithms.

4.5 Improvements on the distance function

Based on the previous results we choose Stay Point Detection as the algorithm on which to improve. Stay Point Detection uses a distance function, which in the case of GPS, represents the geographical distance between two positions. We used the same distance function for our WiFi remote-positioning data set. However, the circumstances are different from the GPS case. Possible positions are few and they represent the locations of the sensors.

The geographical distance is not the best representation of the length people need to walk to get from one place to another. Consider two sensors with a building between them, pedestrians cannot move in a straight line between the two sensors, instead they need to go around the building.

Using the street network and replacing distance with the length of the shortest path from one sensor to another is an alternative. However, street maps are not always available. Consider sensors placed inside buildings. When street maps are available, they lack detail, for instance, popular paths may be hidden between buildings or through parks. Pedestrians may prefer them in order to avoid long walks. Furthermore, street maps do not reflect pedestrian preference. A longer path between two sensors might be preferred if it offers a more pleasurable walking experience, caused by placement of shops or green spaces.

Our goal is to obtain a distance function based on the data itself. We can take a WiFi remote-positioning data set and count the numbers of detections of a device at one sensor followed by detections at another. If we do this for all sensor pairs, we obtain a set of values that represents the closeness of sensors, as it appears from the movements or simultaneous detections of people. It comes naturally that sensors close to each other have a high number of consecutive detections. In contrast, it is unlikely that sensors placed further away would have consecutive detections because sensors in between them could detect the device.

To better understand this closeness value, it is simpler to consider a higher scale. Say we have sensors in two cities. The number of consecutive detections at a pair of sensors where one is in one city and one in the other is expected to be very small. However, if we place sensors at the airports connecting these two cities, we would observe a high number of consecutive detections. We claim that from a transportation standpoint, airports are "close" to each other as detections at the two of them represent a move and no intermediate detections are recorded during the flight.

Extracting these closeness values does not require the use of geographical positioning or any topological information. It uses only the positioning data set. This makes it possible to apply this distance function for any positioning data set, even if the positions of sensors are not known.

WiFi remote-positioning applications have a limited set of sensors. This, in turn, means there are a limited set of closeness values, $\binom{n}{2}$, where n is the number of sensors. We can model a sensor closeness graph, where the sensors are nodes and these closeness values are the weight of the edges between them.

Sensor Neighborhood Graphs

We define multiple Sensor Closeness Graphs. Let $SCG = (S, E)$ be a sensor closeness graph where each node S is a sensor and each edge E has a weight equal to the closeness between the two sensors at its ends.

For each SCG we define a set of sensor neighborhood graphs (SNG). These graphs have the same nodes as the SCG but contain only the edges that have a weight higher than a set ε threshold. An SNG represents sensors close enough to each other to be considered neighbors. With the sensor neighborhood graphs we can better understand and validate the closeness functions as well as make computations more efficient. The distances only need to be calculated once and the verification can be made whether the edge exists in the graph or not.

We have identified the following as distance functions that can be used to build sensor closeness graphs:

Consecutive detections (CON)

Given the set of detections for a device ordered by time, and sensor, we define consecutive detections as two detections of a device, one at one sensor, and one at another. We do not consider how much time passes between detections; we only keep the restriction that they must be consecutive in the set. It comes naturally that sensors close to each other have many such consecutive detections, while those that are far have few. The few consecutive detections at sensors placed far apart can be caused by lost frames at the sensors in between them or gaps in transmission period.

We count the number of consecutive detections for each sensor pair and use the resulting value as the weight of the edge between the two sensors forming the pair. The edges and counting process are formally described in the formulas at 4.2. Here, $C_{CON}(s_i, s_j)$ represents the set of detections recorded at sensor s_j after detections at sensor s_i and E_{CON} represents the sensor closeness graph based on consecutive detections.

We remind the reader that the notations have been defined in Chapter 2.

$$\begin{aligned} C_{CON}(s_i, s_j) = \{k | k \in [1, R]; \exists \lambda_k, \lambda_{k+1} \in \overline{\Lambda}; \\ \lambda_k^S = s_i; \lambda_{k+1}^S = s_j; \lambda_k^D = \lambda_{k+1}^D\} \\ E_{CON} = \{(s_i, s_j, w_{ij}) | \forall s_i, s_j \in S; w_{ij} = |C_{CON}(s_i, s_j)|\} \end{aligned} \tag{4.2}$$

Simultaneous detections (SIM)

In WiFi remote-positioning data sets we can have consecutive detections with the same, or different time stamps. We consider two detections to be simultaneous if they have the same time stamp. This means the device is in range of both sensors at the same time. These detections are counted similarly to the consecutive ones, but they have the added restriction of having to occur

at the same time. We note that it is possible for detections of different frames to have the same time stamp. Building graphs based on simultaneous detections is formalized in formulas 4.3. Here, $C_{SIM}(s_i, s_j)$ represents the set of detections recorded at sensor s_j at the same time as sensor s_i and E_{SIM} represents the sensor closeness graph based on simultaneous detections.

$$\begin{aligned} C_{SIM}(s_i, s_j) &= \{k | k \in C_{CON}(s_i, s_j); \lambda_k^T = \lambda_{k+1}^T\} \\ E_{SIM} &= \{(s_i, s_j, w_{ij}) | \forall s_i, s_j \in S; w_{ij} = |C_{SIM}(s_i, s_j)|\} \end{aligned} \tag{4.3}$$

Simultaneous detections validated with frame sequence number (SEQ)

We record detections with a time resolution of one second. However, the WiFi-frame-transmission frequency is much higher. This means that even if we consider detections to happen simultaneously, this is not a guarantee that the two sensors receive the same frame. Most WiFi frames contain a sequence number, making it simple to distinguish frames. In the Assen 2016 data set we recorded these sequence numbers along with every detection. We use λ^N to denote the sequence number recorded with a detection. For this closeness distance we keep all the restrictions from the previous ones and add the restrictions that the detections should be of the same frame, based on the sequence number, as can be observed in formulas 4.4. Here, $C_{SEQ}(s_i, s_j)$ represents the set of detections recorded at sensor s_j at the same time as sensor s_i, confirmed by the sequence number, and E_{SEQ} represents the sensor closeness graph based on simultaneous detections validated with the frame sequence number.

$$\begin{aligned} C_{SEQ}(s_i, s_j) &= \{k | k \in C_{SIM}(s_i, s_j); \lambda_k^N = \lambda_{k+1}^N\} \\ E_{SEQ} &= \{(s_i, s_j, w_{ij}) | \forall s_i, s_j \in S; w_{ij} = |C_{SEQ}(s_i, s_j)|\} \end{aligned} \tag{4.4}$$

Based on our formulas we can conclude that $C_{SEQ} \subseteq C_{SIM} \subseteq C_{CON}$. This translates in smaller weights for E_{SIM} compared to E_{CON} and even smaller ones for E_{SEQ}. Because the weights are smaller it is also more likely that edges have weights of zero, having a clearer separation between sensors that are close and those that are far.

Geographical Distance (DIS)

To better compare the distance functions, we model the geographical distance between the sensors in a similar manner to the three distance functions presented above. We build a sensor distance graph where the weight of an edge represents the geographical distance between the sensors. From this graph we can build *SNG* graphs by keeping only the edges where the weight is smaller than ε. The formula representing the edges for this sensor distance graph is 4.5.

$$E_{DIS} = \{(s_i, s_j, w_{ij}) | \forall s_i, s_j \in S; w_{ij} = \text{GPSdistance}(s_i, s_j)\} \quad (4.5)$$

By having the *SNG* we can modify the Stay Point detection algorithm so that it does not have to calculate the distance every time. Instead, given two detections it verifies if the two sensors that recorded the detections are connected in the *SNG*. If they are connected, it considers the person has not moved and if the opposite is true, the person has moved, and the pivot is updated to the new sensor.

4.6 Improvement Analysis

We build the graphs as described in the previous section based on the Assen 2016 data set. To have a better understanding on how similar these graphs are we look at the *SNG*s corresponding to each of them. We start with *SNG*s that contain zero edges. We then select appropriate values for the ε threshold so that we add one edge at a time to the *SNG*s. Finally, the *SNG*s will contain all edges, forming complete graphs. We remind the reader that sensor neighborhood graphs contain edges with weights larger than ε for the sensor closeness graphs and smaller than ε for the sensor distance graph.

For each pair of distance function and every number of edges in the *SNG*s we calculate the percentage of edges in common. The results are presented in Figure 4.4. The differences appear because depending on the distance function the order of edge weights differs from one graph to the other. To better understand the results, we added two graphs whose edge weights are chosen to be random. We compare the two random graphs using the same technique as with *SNG*s, by adding one edge at a time to both. As we can observe for Rand-Rand, the percentage of edges in common grows close to linearly with the number of edges.

The most similar *SNG*s are generated using the consecutive and simultaneous distance functions (CON-SIM). Simultaneous detections represent a third of consecutive ones. The graph generated using the sequence number distance function is the closest to the one based on the geographical distance (DIS-SEQ). This indicates that the distance function which uses the sequence numbers is our best replacement for geographical distance.

To understand the effects of different distance functions on our data set we selected a sample of 100 devices and used the modified Stay Point Detection algorithm that takes *SNG* instead of distance to partition traces into stops and

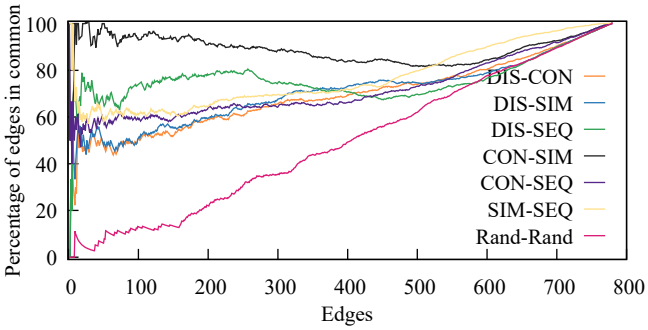


Figure 4.4: Comparing graphs by counting edges in common

movements. We selected appropriate values for ε so that we add one edge at a time to the *SNG* and we executed the algorithm using all the resulting graphs for each distance function. The number of movements is presented in Figure 4.5 and the total duration of stays is presented in Figure 4.6.

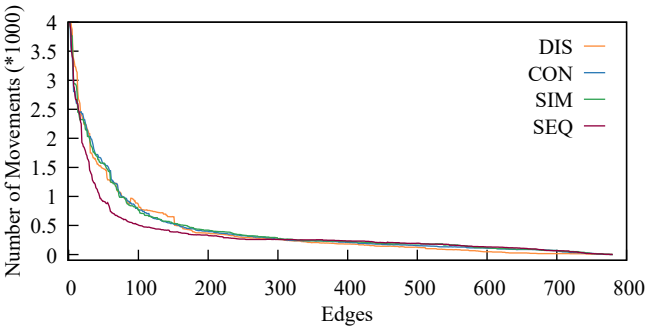


Figure 4.5: Comparing graphs based on the number of movements

With an increase in the number of edges we can see that more movement is identified as a stop period. When the *SNG* is full no movement can be identified. This is normal, as all sensors would join to act as one, very large sensor.

The distance function based on the sequence numbers stands out in both the number of movements and the stay durations.

To further test the performance of the new distance functions, we manually selected two groups of 100 devices from our data set. One with devices

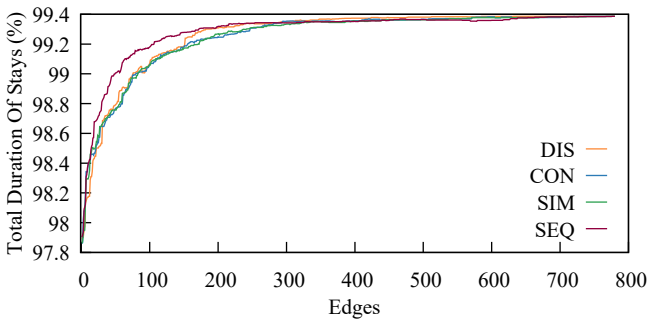


Figure 4.6: Comparing graphs based on the total duration of stays

we identified as mobile (M) and one with devices identified as static (S). We executed Stay Point Detection, using the four different distance functions, on these two groups of devices to verify if the devices are correctly identified as static or mobile respectively. A device is considered mobile if it contains at least one move period, otherwise it is static. As we can see in Figure 4.7 regardless of the distance function, when we add more edges to the *SNG* fewer devices are identified as mobile in either group. The static devices are mislabeled when we have few edges in the sensor graph because any detections at multiple sensors not connected by an edge is considered movement of the device. With many edges in *SNG* movements are no longer identified.

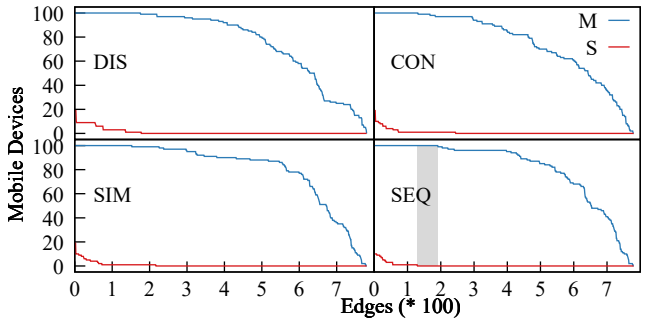


Figure 4.7: Comparing graphs using static and mobile devices

Only in the case of the distance function based on the sequence number do

we have perfect labeling of both groups. The values of ε for which we have perfect labeling are marked in Figure 4.7 with a gray bar. For the other distances, the ε threshold can be set to perfectly detect moving devices, or perfectly detect static devices, but not both.

To confirm that the distance function based on sequence number brings an improvement over the geographical distance used by Stay Point Detection we compare the results after applying the algorithm on the traces for the nine devices for which we have ground truth. The F1 scores when comparing the results from the two Stay Point Detection methods as well as the "Perfect" solutions are presented in Figure 4.8. Although the improved version of Stay Point Detections does not offer a higher F1 score for every device, the average is higher. The number of edges in the *SNG* for the improved version is set to be the smallest one for which the algorithm offers perfect accuracy in labeling mobile and static devices as was presented in Figure 4.7.

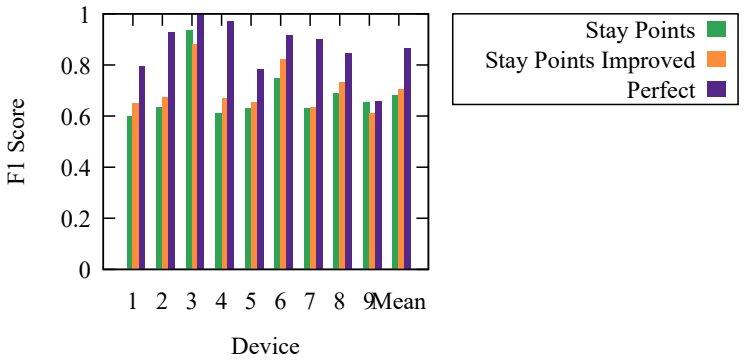


Figure 4.8: Comparing algorithms

4.7 Summary

The number of positions in a crowd dynamics monitoring data set is not a good estimate of the ability to use this data to model crowd dynamics. A person sitting can be detected many times, yet none of these detections add information when we try to model flows.

Crowd dynamics represent the sum of the movement of the individuals inside the crowd. The more movement we can describe, the more realistic our

crowd model would become. As such, movements are the summary of a data set of positions.

The number of movements is a good estimate of the amount of information that we can find inside a data set of positions. But detecting a device at two different positions does not mean it moved. The low positional accuracy and the high frame loss generate many cases where traces based on the positional data show movements where there are none.

We have identified three algorithms that find movements based on the positional data. These algorithms were designed to work for GPS where a similar circular-movement behavior appeared but on a much smaller scale. We compared the three algorithms by measuring the accuracy of splitting a trace in periods of stops and moves and determined that the stay point detection algorithm performed the best. We then improved the algorithm by replacing the distance function with a one calculated based on the data set of positions.

Our improvement manages to bring a slightly higher accuracy, but it does this without requiring as input the position of the sensors. Remote positioning platforms built on top of existing WiFi networks may not have information on the position of the sensors.

More importantly, we showed that even an algorithm that would perfectly label each detection as one where the device is moving, or it is stopped cannot achieve perfect accuracy. It would have a mean F1 score of 0.86. This is because the time between detections is generally too large. Furthermore, our best implementation has achieved an average F1 score of only 0.7. This shows how difficult it can be to extract useful information out of WiFi remote-positioning data.

CHAPTER 5

Sensor density and placement

New, innovative applications make use of data stores to extract information on complex systems. With more data, the accuracy and correctness of the extracted information can be increased. However, gathering more data is not always a trivial task.

We have shown that data obtained from WiFi remote-positioning systems is sparse, limiting the amount of information that can be extracted. Most devices have few detections, others have large gaps between detections, while others may not be detected at all. Without any control on the target devices, the only way to increase the amount of data, and in turn information, produced through WiFi remote positioning is by increasing the number of sensors. More precisely, by increasing the density of sensors. Adding more sensors while also extending the interest area brings more data, however this data cannot be used to answer more specific questions about the original interest space.

Studying the effect that the density of sensors has over the sparsity of detections, and more importantly on the amount of information that can be extracted, is vital. With more information we can improve crowd-dynamics models, and if we can determine an optimal density of sensors, we can lower the cost of monitoring platforms. The sensors represent the main factor in the cost of a crowd-dynamics monitoring system based on WiFi remote positioning. Lowering sensor density makes crowd-dynamics monitoring platforms more accessible.

Low sensor density is especially important for projects that need to cover large areas, such as entire cities. We are not aware of any deployment of WiFi remote-positioning systems that cover an entire city or of any research that concentrates on lowering the density of sensors for WiFi remote-positioning applications.

Sensors are identical, meaning the only property that changes from one sensor to the other is their position. The number of sensors and their position

are closely linked properties, and as such can be addressed at the same time. As we explore the effects of sensor density, we take positions into account and try to understand what makes a "good" sensor placement.

5.1 Contributions

We study the effects that the sensor density has on the amount of information that can be extracted from WiFi remote-positioning data sets. The amount of information is closely related to the number of stops and moves that can be extracted from the data set. This is because, stops and moves, as they are described in the previous chapter are the summary of a trace.

To thoroughly study the effects of changing the density of sensors we require a large ground-truth data set containing data on many individuals. It is arguably impossible to gather such a ground-truth data set. Instead **we simulate the movement of individuals and the detections they would trigger**. We make use of a real and a synthetic map for the simulated movement and validate the results with a smaller real-life data set for which we obtained ground truth.

To obtain different densities of sensors we perform our analysis on subsets of a data set, subsets that contain detections from a selection of sensors. This permits us to compare data of the same individuals, same movements, and eliminates the risk of random variations affecting our results.

We explore our measurements and explain them through a deeper analysis on sensor placement and the effect of unique detections. **Finally, we combine our findings into a short guide on sensor placement for WiFi remote-positioning platforms**.

5.2 Related Work

In most positioning systems sensor density influences positional accuracy. Consider GPS, where accurate positioning can be obtained only when signals from more than three satellites are received [122] and accuracy increases with the number of satellites [123]. However, this increase is limited [124] and after a threshold is reached, receiving signals from more satellites offers no measurable increase in positioning accuracy. Increasing positional accuracy by increasing the number of sensors does not work for WiFi remote-positioning systems. We have shown in previous chapters, that the sparsity of simultaneous detections

and the high variation of the RSSI, limit positioning accuracy to be dependent on only one sensor.

Gathering crowd-dynamics data using remote-positioning based on communication protocols is a relatively new technique, so there are limits to how far the state of the art has gone. A topic that has not been sufficiently addressed is the effect of the density of sensors and their position on the sparsity of data and the amount of information that can be extracted from it. The closest work that deals with this problem is [125], but the scope of their work is limited to determining optimal placement of Bluetooth sensors on a linear path, in practice, a highway. This solution, although interesting, is not enough when we consider crowd dynamics for complex street networks.

There is the assumption that having more sensors, and with them more data collected, means an increase in the amount of information that can be extracted from this data. However, the number of sensors is a main factor to the financial cost of a monitoring platform and because of this, most research is focused on minimizing the number of sensors while maintaining an acceptable level of information that can be extracted from the data which the platform provides.

One solution applies to measurements that are representative of not just the precise location of the sensor, but the area around it. This is the case in [126] where they show how to determine the optimal number of sensors for structural health-monitoring systems. Defects in a building affect and can be detected over large areas. Similarly, the work of [127] demonstrates the use of a technique of minimizing the number of sensors for water-supply networks. This is based on the fact that pollution distributes itself uniformly through air or water. As such, making punctual measurements is representative of large areas around it.

Unlike the cases where the measured variable is continuous across space, crowd-dynamics monitoring platforms generate complex, interdependent data. This data can be aggregated in order to obtain continuous values, but this limits the use cases. An example would be aggregating raw detections to discover the density of people over space. However, more interesting uses for crowd-dynamics monitoring data requires individual traces. Individual traces cannot be modeled such that they exhibit continuous variation over space. A simple example where trace data can be used is to determine the speed of crowd flows, which is an average of the speed of individuals it represents. Different crowds may have different speeds which may not be related. Another example would be the likelihood of landmarks to be visited in a given order.

Minimizing the number of sensors has been studied extensively for a more general case, the one of sensor networks. The solutions that deal with generic sensors and sensor networks address mainly the coverage that the sensors offer

in both a 2D environment [128] as well as a 3D one [129]. The latter offers a technique to place sensors on complex geographical landscapes such as hills or mountains. Another goal can be to try and maximize the exposure that the monitored elements have using a specific sensor placement [130]. We have conducted previous research on this topic [131] considering cases where sensors are mobile.

We argue that we cannot assume that coverage or exposure are the only factors that influence the amount of information that can be extracted from WiFi remote-positioning data sets. Take the case of stops and moves, two well-placed sensors, one at the workplace and one at home can offer all the required data to determine the stop and move periods for an individual, if the movements are limited to these two locations. Time constraints could even reveal clues about what path was taken between the two. These results can be applied on our work while keeping the goal of extracting relevant movement information and allowing us to remove the constraint of having full coverage. This could permit lower sensor density while maintaining acceptable results.

5.3 Procedure

We showed in Chapter 2 that changing sensing parameters such as the monitoring channel does not affect the data generated by WiFi remote-positioning systems. Other sensor software modifications, such as those that allow more data to be extracted from the WiFi frames, bring privacy concerns on which we are not willing to compromise.

The only changes that we can bring to WiFi remote-positioning systems, other than changing the recording parameters of the sensors, pertain to the number and position of sensors. Of course, if we use more sensors to gather data on a larger area, we would gain more data and more information. What we are interested in is changing the number of sensors while covering the same area. This translates into analyzing the effects of sensor density.

We aim to study the effects that sensor density has on the amount of information that can be extracted from WiFi remote-positioning data sets. We expect the number of raw detections to grow almost linearly with the density of sensors. However, we do not know what effect this growth has on the amount of information that can be extracted.

The amount of information can be increased by improving positional accuracy; however, improving positional accuracy is not possible. Increasing it requires the rate of simultaneous detections to be significantly increased. As-

suming enough simultaneous detections, we can use trilateration to calculate positions with higher accuracy. Considering the high rate of frame loss as well as the highly unpredictable nature of outdoor sensing, this assumption requires a sensor density which is not realistic. Instead of addressing the positional accuracy, we focus on the information that can be extracted while maintaining positional accuracy obtained without trilateration.

We explore the effect of sensor density in order to achieve two goals. First, we want to determine if we can add more sensors to an existing WiFi remote-positioning platform in order to increase the amount of information that can be extracted. We showed in the previous chapters that the amount of data and information generated by WiFi remote positioning is underwhelming. Secondly, if we can lower sensor density while maintaining the same level of extracted information, we can lower the financial cost of WiFi remote-positioning platforms.

The amount of information output by crowd-dynamics monitoring platforms can be correlated with the number of stop and moves that can be identified using the positioning data. Stop and moves are a short, simple representation of the dynamics of an individual and can be used in modeling crowd dynamics.

Algorithms that extract stop and moves do not offer perfect labeling and this is an impediment if we want to make comparisons using the numbers of stops and moves. Low positional accuracy during a stop period can be labeled as movement. Instead of using the number of stops and moves to represent the amount of information, we use the accuracy of correctly labeling periods as stops or moves given positioning data and ground truth. High accuracy translates in the ability to extract more correct stops and moves and in turn more information. By using accuracy as a comparison metric, we can determine the amount of information without being affected by the bias introduced by the stop and move algorithm.

To extract stops and moves we use the **Stay Point Detection** algorithm. We showed in the previous chapter that this algorithm is the best performing for WiFi remote-positioning data sets. For the analysis in this chapter we use a maximum distance threshold of 220m, maximum movement period between two detections of one hour and minimum stay period of 20 minutes. We determined these values empirically when we studied the performance of the algorithm.

To measure accuracy, we use the F1 score to compare the periods of stop and moves generated by the stay point detection algorithm with the stop and moves from the ground truth. The F1 score gives equal importance to precision and recall and this is ideal when one class (such as the total stop time) dominates

the data set. Because we have only two labels, we can simplify the accuracy calculations by considering moves to be positives and stops to be negatives. We then compare the labels for every one second interval and count the number of true positives (TP), false negatives (TN) and false positives (FP). We use these values to calculate the F1 score using Eq. 5.1.

$$F1Score = \frac{2 * TP}{2 * TP + FP + FN} \tag{5.1}$$

Comparing the accuracy of extracting stop and moves from data sets gathered using different sensor densities would require many data gathering experiments. During these experiments, we would have to obtain large amounts of ground truth. It is not feasible to collect crowd-dynamics ground truth on a large scale. Furthermore, for each sensor density, we would need to conduct multiple data-gathering experiments in order to mitigate for randomness in crowd dynamics. We must limit our analysis to smaller data sets (for which ground truth can be obtained) and simulations.

To conduct our analysis, given a small real-world data set and while avoiding random phenomena, we propose varying the sensor density by selecting multiple subsets from a main positioning data set. Each data subset contains only detections recorded at a subset of sensors. This allows us to compare the same crowd dynamics given different sensor densities while eliminating random phenomena that could appear in one set but not the other. Let $S_A \subset S$ be a subset of sensors. The subset of detections Λ_A at sensors S_A is extracted using equation 5.2. Here, λ_i represents a detection from the entire set of detections Λ and λ_i^S represents the sensor that recorded the detection.

$$\Lambda_A = \{\lambda_i \in \Lambda; \forall i | \lambda_i^S \in S_A\} \tag{5.2}$$

Extracting the subset using this method results in a data subset which is equivalent to the one gathered using only the selected sensors. This is true because sensors do not interfere with each other. Adding one sensor does not influence the data set gathered by the existing ones.

Given a data set gathered using N sensors we can extract data subsets having detections recorded at a number of sensors k, with $k \in [1, N]$. The problem is that for each value of k we have $\binom{N}{k}$ possibilities as to which k sensors to select. If we were to analyze all possible combination of sensors given all values for k, we would have to process $2^N - 1$ data subsets. This value is obtained using Eq. 5.3.

$$\sum_{k=1}^{N} \binom{N}{k} = \sum_{k=1}^{N} \frac{N!}{k!(N-k!)} = 2^N - 1 \tag{5.3}$$

With so many possible choices of sensor subsets, we cannot analyze all possible data subsets. However, we need to analyze only enough so that we have an overview of the accuracy values. We propose the following scheme: for each value of k, we make 10 choices at random, along with one choice so that we obtain the highest accuracy, an upper bound, and one to obtain the lowest accuracy, a lower bound. This results in 12 sensor samples for each k value. We make the 10 random choices in order to show what information can be extracted if the location of sensors is left to chance.

Finding the real lower and upper bound for each k can only be done by analyzing all $\binom{N}{k}$ choices of sensors, which quickly becomes infeasible for many reasonable combinations of N and k. Instead, we propose a greedy algorithm to find the selection of sensors for the lower and upper bound by incrementally selecting sensors for consecutive values of k. The goal of the algorithm is to obtain data subsets that can be processed into periods of stops and moves having a minimal or maximal accuracy. Accuracy which would be similar to the one for the real lower and upper bounds.

The greedy Algorithm 2 is initialized by selecting a pair of sensors that offer the lowest, respectively highest accuracy for $k = 2$. We do not start at $k = 1$ because one sensor can detect no movement, meaning they would all have the same F1 score of zero and the choice of sensor would be left to chance. Starting at $k = 2$ ensures that both sensors are specifically selected to minimize or maximize accuracy. After we discover the two sensors, we increase k one step at a time and for each step we select another sensor so that when we add it to the previous set we would minimize, respectively maximize the accuracy. For each k we need to go through all sensors that are not already selected. Algorithm 2 presents the greedy implementation on how to find the upper bound. An algorithm that finds the lower bound is symmetric to this one. The *max* variable

is changed with *min* and so are the signs inside the if clauses.

Data: Λ, S, N
Result: S_A
$S_A = \emptyset$;
$max = 0$;
for $s_i, s_j \in S; i \neq j$ **do**
  $acc = F1(StayPointDetection(\Lambda[\{s_i, s_j\}]), GroundTruth)$;
  if $max < acc$ **then**
    $acc = max$;
    $S_A = \{s_i, s_j\}$;
  end
end
for k=3 *to* N **do**
  for $s_i \in S; s_i \notin S_A$ **do**
    $acc = F1(StayPointDetection(\Lambda[\{S_A, s_i\}]), GroundTruth)$;
    if $max < acc$ **then**
      $acc = max$;
      $s_a = s_i$;
    end
  end
  $S_A = \{S_A, s_a\}$;
end

Algorithm 2: Finding upper bound subsets using a greedy approach (analogous for lower bound)

Using the greedy method we have to run the **Stay Point Detection** algorithm only on $N(N-1)/2$ data subsets. Although still large (the processing took several weeks), analysis on these many data subsets is feasible, compared to what is required to test all $2^N - 1$ subsets. We believe that analysis based on data subsets obtained with the greedy method is sufficient to support our conclusions.

We apply this procedure in order to determine what is the dependency between the accuracy of labeling stops and moves and the sensor density. Furthermore, **by using the greedy algorithm to build the data subsets for the lower and upper bounds of the accuracy, one sensor at a time, we obtain two ordered sets of sensors**. We can analyze these ordered sets in order to gain insight on the effects of sensor positioning.

5.4 WiFi remote-positioning data sets

To determine the effect of sensor density on the accuracy of labeling stop and move periods we analyze three different data sets. Two of these data sets are simulated, one using a grid map, the other using a real map. We use simulated data sets because this way we can provide large amounts of ground truth compared to real-world data-gathering experiments. The real map is the city center of Assen, where we performed most of our data-gathering experiments. This allows us to compare the results for the simulated data set with the real-world one, obtained in Assen in 2016.

5.4.1 Simulated data on grid map

We use a grid map in order to ensure that the shape of a specific city does not affect our results. The regularity of the grid structure prevents any bias that may be added when using the map of a specific city.

We built a map of roads organized as a grid. Every intersection has two roads connected at right angles. The distance between two intersections is fixed and set to 100m. We chose this value because it allows every street sector to have between 2 to 10 houses, which is realistic, and because it simplifies the placement of sensors. The entire street map will be 900m by 900m and will have a total of 100 intersections. We chose the distances so that if we place a sensor at every intersection, the entire area will be completely covered. Although this map is only a mock-up, some real cities, especially in the USA, share this grid structure.

The grid map, Figure 5.1, shows sensor locations as small white circles. The large white circle has an 100m radius, representative of the advertised distance of the WiFi range. The black lines connecting the small white circles represent the streets and are the only areas that pedestrians can use.

We built simulated data sets using sensor ranges of 50m, 100m and 150m. Other than the detection ranges all configurations are identical and the simulations share the same movements. In Subsection 5.5.3 we show how the results can differ depending on the sensing range. However, the differences are not substantial and for our main analysis we use the range of 100m.

5.4.2 Simulated data on Assen map

Several of our data gathering experiments took place in the city of Assen, The Netherlands. The map of Assen as well as the locations where our sensors were

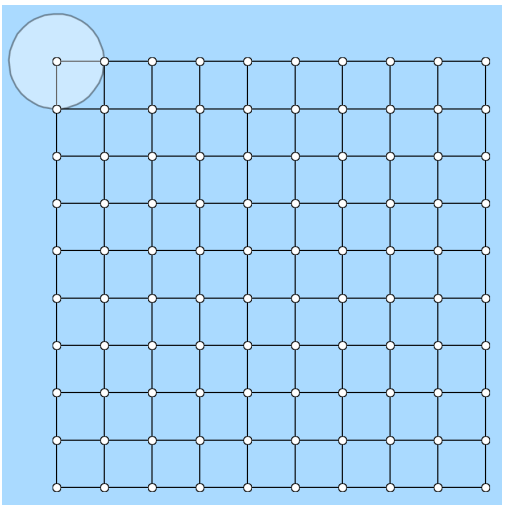


Figure 5.1: Grid map, sensor placement

placed during the experiments are shown in Figure 5.2. Here, the streets are white, and the sensors are spread in the important areas of the city center. A few sensors were placed further away, and they are not represented in this picture. During our simulation pedestrians make use exclusively of the street network.

In order to obtain comparable results with the real-world data sets, we maintain the same sensor placement for the simulated data set. Having comparable setups allows us to contrast the real-life and simulated results and to determine if our simulation has fundamental flaws.

5.4.3 Real-world data - Assen map

In order to verify our results, we compare them with a data set obtained from one of our data-gathering experiments. During the Assen 2016 data-gathering experiment we formed a group of four people and went in the festival area carrying nine mobile devices, tablets and smartphones. We stayed as a group and made multiple stops during several hours for each of the two days in which we visited Assen and the festival area.

We made sure to record our movements both with a GPS tracker and by taking notes in case of errors in the GPS data. These notes and the GPS data

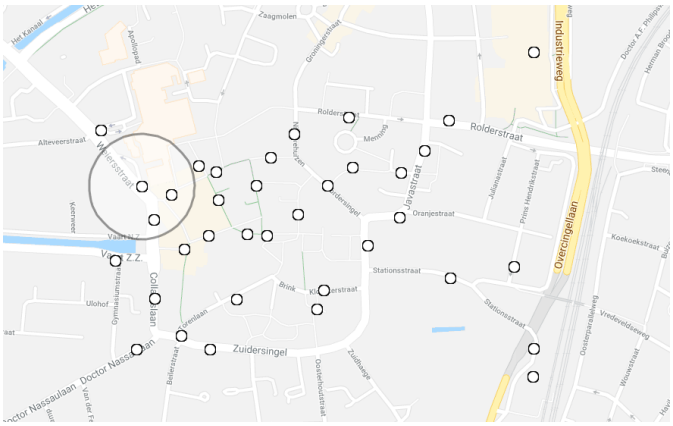


Figure 5.2: Assen map

serve as ground truth for our movements.

5.4.4 Simulating movements and detections

Using the two maps (grid and Assen city center) we generate two simulated WiFi remote-positioning data sets. By using simulated data, we can control factors such as packet loss or movement speed. It is not feasible to perform such experiments in real life at a large scale.

Given a street map we generate the walking paths for 10 people. We assume the individuals are randomly walking (a popular technique for simulating human movement). To generate a path, we select a random position from the street map and set it as the starting point. We then select a new position, more than 300m away from the first, and have the individual take the shortest path between the two points. The process is repeated until the simulation reaches a predetermined end time, of 24 hours. We model stop and move periods for intervals of one, resulting in 240 periods. The first hour is a moving interval, followed by a stop hour, then, another moving interval, simulating in total 24 hours of data for each of the 10 people. When a stop period starts the individual maintains its current position regardless if it reached its destination or not. The movement speed is set to about 2m/s (walking speed). The specifics of these simulations are similar to the simulation we generated for Chapter 4.

We record detections every second at the sensors that are in range of the

device as given by the path that the individual carrying it takes. The sensors are considered ideal, with a detection range of 50m, 100m and 150m (for the main analysis we use 100m, the advertised range for WiFi devices) and with a detection area in the shape of a disc. We chose a disc-shape in order to circumvent random variation introduced by a more realistic, irregular detection area.

The traces will have periods of one hour of constant movement, followed by periods of one hour of stops. We take this set of one-hour periods and use it as ground truth. Because the data is simulated, we can generate the ground truth alongside the data set of detections.

5.5 Analysis

For our analysis, we compare and measure the effect of having different sensor densities for crowd-dynamics monitoring platforms based on WiFi remote-positioning systems. In order to have an unbiased comparison we need data sets generated using different sensor densities that represent the same crowd-dynamics phenomena. To build such data sets, we applied the procedure presented in section 5.3 on the three original data sets presented in 5.4. This procedure allows us to extract multiple data subsets from one WiFi remote-positioning data set with fine control over the number of sensors that will be included in the data subsets.

5.5.1 The effect of sensor density on move and stop labeling

For each of the three original data sets (simulated - grid map, simulated - Assen map, real-world - Assen map) we extract data subsets with the number of sensors k ranging from 1 to N. N is the number of sensors in the original data set (100, 40 and 40 respectively).

For each number of selected sensors k (X-axis) we select 12 data subsets. The 12 data subsets are chosen using three sensor-selection criteria: 10 are randomly selected, the other two are determined using the greedy algorithm in order to obtain a lower and an upper bound. The selection criteria, and the reason for choosing 12 subsets, have been described in detail in Section 5.3.

To compare the data subsets, we use the accuracy of correctly labeling periods of stops and moves with a resolution of one second. In other words, we compare the labels of each one second interval. Periods of stops and moves can have different lengths. Because of this we use the F1 score, which balances

precision and recall. This makes the F1 score sensitive even in scenarios where one class dominates the data set. The F1 score is also the metric used for the greedy algorithm in order to discover the lower and upper bounds.

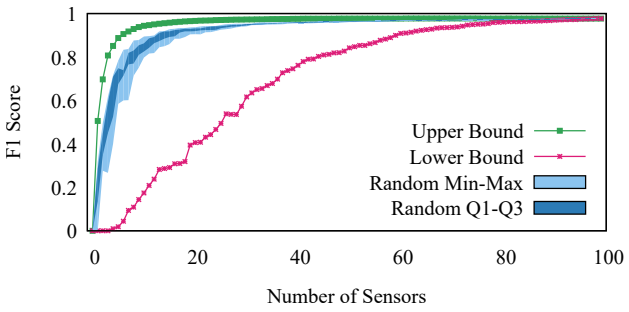
The graphs in Figure 5.3 show how the variation in the number of sensors (we maintain the same interest area, meaning the number of sensors is equivalent to sensor density) affects the F1 score. The x-axis represents the number of sensors contributing with detections to the data subsets. The y-axis represents the F1 score obtained by comparing the stop and moves labeled by applying the stay point detection algorithm on the data subset, with the ground truth.

Each of the figures represents comparisons based on different original data sets. Figure 5.3a represents data subsets obtained from the simulated data set using the grid map. Figure 5.3b represents data subsets obtained from the simulated data set using the Assen city center map. Finally, Figure 5.3c represents data subsets obtained from the real-world data gathered in the city of Assen in 2016.

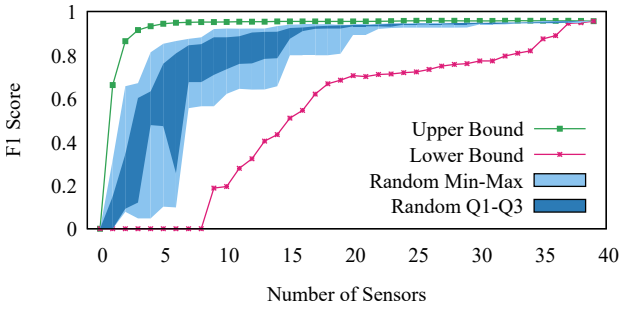
The upper bound is represented with a green line while the lower bound is represented with a red one. Displaying a line for each of the 10 random sensor selections would make the graphs unreadable. Instead, we display a representative area with light and dark blue. Light blue determines the area between the minimum and maximal F1 score values from the 10 randomly selected data subsets. Dark blue is used to represent the area between the first and third quantiles of the 10 F1 scores.

The analysis for all three original data sets shows similar correlations between the number of sensors and the F1 scores. The upper bound grows fast and reaches a maximum given only few sensors. The lower bound grows close to linearly with the number of sensors. The random selections start by growing fast and then continue with a slower, almost linear growth, reaching the peak only when all sensors are included.

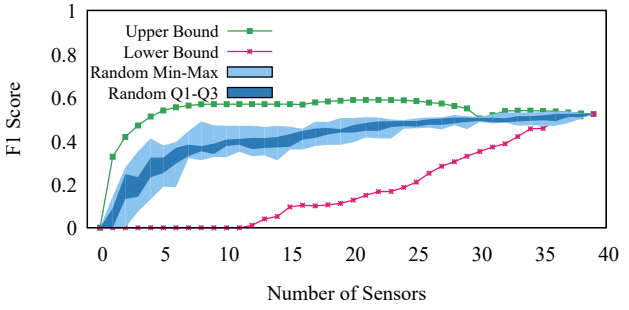
The **upper bound** reaches maximum accuracy given data subsets with just few sensors. The maximal value is reached with 10 sensors for the grid map and 5 for the Assen map. Adding data from more sensors no longer improves the F1 scores. This means more data does not translate into new, or more accurate information. More so, in the real data set, when we reach 30 selected sensors, the F1 score value starts to drop slightly. This means that the data from some sensors, combined with the data from the sensors that were already added makes the algorithm identify movements during stop periods. This can be explained by detections where the real distance between the device and the sensor is exceedingly large. The position recorded for the device is far from the real position, possibly making it seem like it moved. Furthermore, these



(a) Grid map/simulated data



(b) Assen map/simulated data



(c) Assen map/real data

Figure 5.3: Accuracy of detecting stops and moves

low-accuracy detections are common, and usually indistinguishable from the others.

We calculated that 12-15 sensors per square kilometer is sufficient to extract all available crowd dynamics information. We reached this value by comparing the size of the interest area ($0.81km^2$ and $0.33km^2$) with the minimal number of sensors required to reach the maximum F1 score during our analysis (10 and 5).

The F1 scores for the real-world data set are much smaller compared to the simulated ones. The differences can be attributed to lost frames and large periods between detections as well as low positioning accuracy given by irregular detection areas. In contrast, the simulations assume frames are received every second and the detection areas are perfect discs. Even so, with the maximal F1 score being much lower, the slopes for the real data set are akin to the ones we obtained for our simulated data sets. Adding detections with low positioning accuracy can make the algorithm incorrectly label stop periods as moves. The stay point detection algorithm is not sensitive to low accuracy positions (in our case, sensors need to be more than 220m apart to consider a movement), however, if the distance between a sensor and the detected device is larger than 110m we can have mislabeling.

We argue that further increase in the number of sensors (beyond the values represented in our analysis) would not bring any significant improvements of the F1 score. We chose the maximum number of sensors for our analysis the moment when we built our simulations and before we performed our data gathering experiment. The values were chosen to obtain full coverage, with considerable overlap between the sensing areas while maintaining a realistic number of sensors. Sensors can be placed on buildings or light posts and these have a set density.

It is possible, although unlikely, that a further increase in the number of sensors can increase the F1 score. This seems especially true if we look at the lower bound, which shows a linear growth. If we were to extrapolate from it, we would conclude that we should have used a larger number of sensors for our original data sets. However, for all three original data sets, the upper bound remains constant between a relatively smaller number of sensors (10 and 5) and all the way to the maximum number. This makes it unlikely that increasing the number of sensors beyond the values of our analysis would provide data sets for which we can do more accurate labeling and obtain higher values for the F1 score.

The **lower bound** grows almost linearly with the number of sensors. However, especially for the real data set, the F1 score can remain at 0 for as many as 11 sensors. This means it is possible to place sensors in such a way that no

movement is detected between them. This can happen for two reasons. Either the sensors are too close, such that a move cannot be reliably distinguished from a stop period. Or, the sensors are so far apart that no device is detected by any two of them given a reasonable time frame (1 hour in our case). We discovered in the previous chapter that labeling move periods longer than one hour, having detections only at the start and end of the movement, lowers the labeling accuracy.

Randomly selecting sensors results in data subsets that show a rapid growth of the F1 score with few sensors. After the initial rise, as the number of sensors increases the F1 score continues to grow, although at a slower pace. The blue area, representing the F1 scores for our random samples is generally closer to the upper bound. This shows that randomly placing sensors has a high chance of providing data sets from which most of the common crowd-dynamics information can be extracted. Furthermore, the small size of the blue area shows that there is small variation given different sensor samplings and that the F1 score values are affected more by the number of sensors rather than their placement.

The large difference between the upper and lower bounds, as well as the distribution of F1 scores for the random sampling of sensors, shows that sensor placement has a significant effect on the amount of information that can be extracted, although, lower than the effect given by the number of sensors. Randomly placing sensors can offer acceptable results. However, in all scenarios peak accuracy is achieved with a far smaller number of sensors when they are carefully selected (upper bound).

5.5.2 Comparing lower and upper bounds and the number of detections per sensor

Because we use the greedy algorithm to determine the sensor selections for the upper and lower bounds, these selections grow incrementally. The sensors for the upper bound at k are included in the sensors selected at $k+1$. This means that **the lower and upper bounds are based on an ordered sets of sensors**. If each sensor adds a fixed amount of information, irrespective of the other sensors, the ordered sets of sensors for the lower and upper bounds should be almost mirrored.

Sensors can also be ordered by the number of detections. A correlation between the ordering from the upper bound and that of the number of detections would mean that the number of detections can be correlated to the amount of extracted information.

We can compare the ordinal association between two ordered sets using the Kendall Tau distance [132]. The Kendall Tau, τ value is defined in Eq. 5.4, where N is the number of sensors and x_i represents the identifier of the i-th sensor from one ordered set while y_i represents the identifier of the i-th sensor from the other ordered set. For Kendall Tau, a value of 1 would mean the ordered sets are identical, a value of -1 would mean that they are mirrored, a value of 0 would mean there is no correlation between the orders in the two sets.

$$\tau = \frac{2}{N(N-1)} \sum_{i<j} sgn(x_i - x_j) sgn(y_i - x_j) \tag{5.4}$$

Using the Kendall Tau distance, we compare: the two ordered sets (from the lower and upper bound) from each of our three original data sets; the upper bound ordered set with the ordered set we obtain by arranging the sensors increasingly by the number of detections. As a reference we use a comparison between two random permutations of sensors. Figure 5.4 shows the Kendall Tau values for these comparisons.

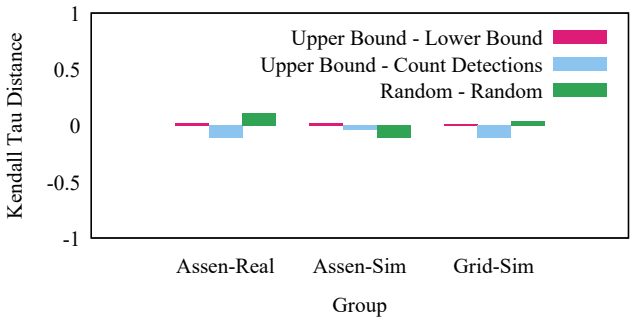


Figure 5.4: Comparison of ordered sets of sensors

All the Kendall Tau values are close to 0, meaning there is no correlation between the ordered sets of sensors, be it the upper/lower bounds sets, random, or between the upper bound and the ordered set based on the number of detections.

Finding no correlation suggests that we can have multiple, completely distinct ordered sets of sensors which offer similar results. When running the greedy algorithm, sensors are chosen based on small differences. This happens because the sensor detection areas overlap, meaning that two or more sensors

record redundant data. Because of these redundancies, it is possible for two different sets of sensors to offer the same information. Consider indexed sensors placed in a line at small distances. Sensors placed on odd positions offer almost the same information as those placed on even positions, although the two sets have no sensor in common and the data sets have no detection in common.

5.5.3 Detection range

We know that the detection area of sensors is irregular and varies in time. With many factors affecting detection range, including dynamics ones such as human bodies or vehicles. As such, it is not feasible to simulate realistic detection areas.

Because we cannot accurately simulate it, sensing distance introduces the main difference between our simulations and real-world data. We study the effect that the sensing distance has on our analysis. To do this we simulated data sets using three sensing distances: 50m, 100m and 150m. We used the 100m sensing distance data set in the previous subsection and for the rest of our analysis. The three data sets are all created based on the grid map and they represent the same movements. This way, we ensure we do not add any noise because of randomness in movements.

For this comparison we calculated the upper and lower bounds of the F1 score of correctly labeling periods of stops and moves given the three sensing distances. The results are presented in Figure 5.5.

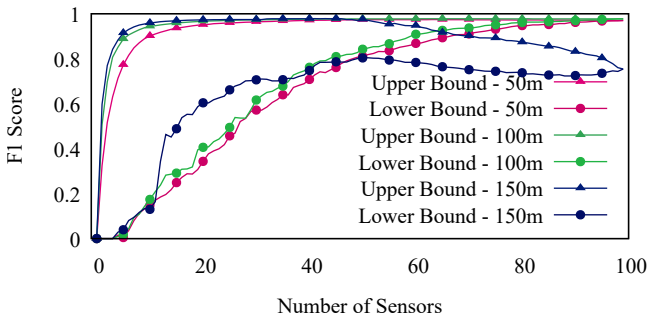


Figure 5.5: Grid map/Simulated - 50m, 100m, 150m comparison

Given few sensors, increasing the sensing distance to 150m increases the accuracy of correctly labeling stops and moves. The improvement is not significant for the upper bound. However, for the lower bound, we can observe a

large difference. The increase in accuracy is due to the larger coverage. Having a larger sensing radius means that we can use fewer sensors to cover a larger area, gather more data and more information.

The opposite is true when the sensing distance is lowered to 50m. The accuracy of correctly labeling stops and moves is slightly lowered.

What we found interesting is that when we have a large detection radius (150m) and the number of sensors is increased, the accuracy drops below the maximum. This happens because we maintain the 220m threshold for the stay point detection algorithm before we consider a movement. With the larger range, the threshold is no longer sufficient. However, increasing the threshold means some short moves can no longer be identified.

The differences introduced by changing the detection range are small and the results from our analysis based on the simulations are on par with the results for the real-world data set. Based on this, we consider that our simulations are accurate enough to be used in order to address differences introduced by sensor density and placement.

5.5.4 Unique detections versus accuracy of stop and move labeling

We are interested to know if the amount of information (represented through the accuracy of labeling stops and moves) that can be extracted from WiFi remote-positioning data, as we vary the number of sensors, can be explained by features such as the number of detections per sensor. A strong correlation between the two would translate in a requirement to obtain more positioning data, in order to be able to extract more information.

In subsection 5.5.2 we have determined that the number of detections does not correlate with the amount of information. We did not find a correlation because neighboring sensors can record the same detections, becoming partially redundant.

Instead of analyzing the raw number of detections per sensor we want to extend our investigation to the number of unique detections. We know that detections recorded at small intervals are recorded at nearby sensors because human walking speed is low. Because of this, we can ignore spatial positions and consider uniqueness only in regards with time.

Definition: *We say a detection is unique if there is no other detection of the same device, at any sensor, in a time interval that starts five minutes before the recording time of the selected detection, and ends five minutes after*. The set of unique detections

Λ_U is defined using equation 5.5. Here, λ_i represents a detection from the set of detections Λ and λ_i^T represents the time at which device λ_i^D was detected. Applying this rule to a set of detections in order to filter the set until it contains only unique detections results in many random choices, based on the order on which the rule is applied. The random choices make it difficult to use the number of unique detections per sensor for our analysis as the numbers can vary depending on chance.

$$\Lambda_U = \{\forall i; \lambda_i \in \Lambda | \nexists j; i > j; \lambda_i^D = \lambda_j^D; |\lambda_i^T - \lambda_j^T| <= 300\} \tag{5.5}$$

The redundancy of detections can be used to represent the sensors as a graph, where an edge represents how much redundancy there is between two sensors. These graphs are akin to the sensor neighborhood graphs which we discussed in Chapter 4. Because of the graph structure, there are many orderings of sensors in which we can apply the uniqueness rule.

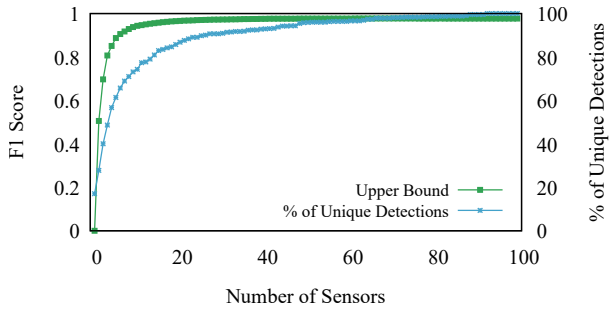
In order to compare the number of unique detections to the accuracy of labeling stops and moves we apply the uniqueness rule using the sensor order obtained by the greedy algorithm for the upper bound. Alternatively, we could have chosen the lower bound. This means that we take all detections for the first sensor, apply the uniqueness rule on them, then we add detections for the next sensor, apply the uniqueness rule again, and so on until all data has been processed.

In Figures 5.6 we plot the F1 score obtained for the upper bound along with the percentage of unique detections as it varies with the number of sensors. To reiterate, the percentage of unique detections is calculated incrementally based on the ordered set of sensors for the upper bound.

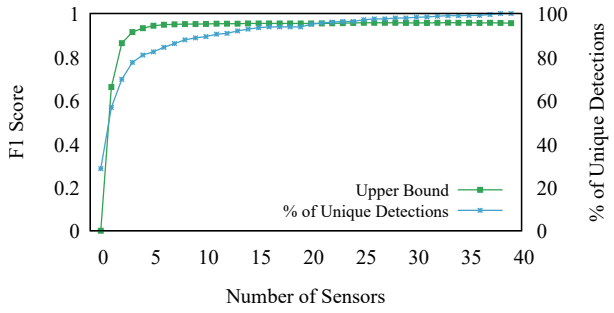
The slope of the F1 score is similar to the slope of unique detections. This is true for the three original data sets. However, there are differences. This means that the number of unique detections is the main factor that determines the amount of information that can be extracted from WiFi remote-positioning data, but other factors exist.

In Figure 5.6c, for the real-world data set, we can observe that the percentage of unique detections grows again after 17 sensors. We require only five sensors to get almost maximum F1 score. The next 12 sensors add no new information and no new unique detections. The number of unique detections does grow given more than 17 sensors; however, the accuracy is not increased, meaning the new unique detections have no positive effect on the accuracy or the amount of information that can be extracted.

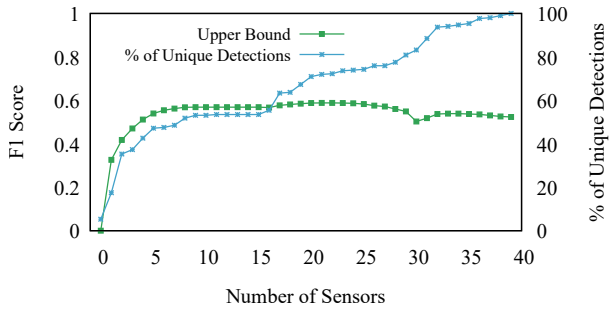
The similarity between the percentage of unique detections and the F1 score



(a) Grid map/simulated data



(b) Assen map/simulated data



(c) Assen map/real data

Figure 5.6: Upper bound vs unique detections

shows the importance of gathering more data. This means that sensors should be first placed in high density areas. Areas where a lot of people are expected, such as shopping areas, stations, parks.

5.5.5 Placement of sensors

So far, we have determined that few sensors are sufficient to obtain high accuracy of labeling stops and moves. Furthermore, we know that gathering more unique positioning data is beneficial.

We want to determine what makes "good" and "bad" sensor placements. The first sensors in the upper bound ordered set have "good" positions because they contribute the most to increasing the accuracy of correctly labeling stops and moves. In contrast, the first sensors for the lower bound ordered set are specifically selected to bring the lowest accuracy.

We extracted the first 10% of sensors from the upper and lower bound ordered sets. We marked the "good" sensors (first 10% of upper bound) with green and the "bad" (first 10 % of lower bound) with red on the sensor maps in Figures 5.7, 5.8 and 5.9.

We observed that it is possible for a sensor to be both in the first 10% from the upper bound and 10% from the lower bound. We labeled these sensors with black. These sensors belong in both groups due to the overlap between detections. When the greedy algorithm is used for the upper bound, these sensors are far from the others, meaning they add unique detections and new information. In the case of the lower bound, they are clustered together with other "bad" sensors, meaning they provide redundant detections and no new information. Having the sensors be "good" or "bad" based on their relationship and small variations explains why the ordered sets from the upper and lower bound are not mirrored and why the Kendall Tau distances between them are close to 0.

From all maps, we can observe that the best practice is to spread the sensors and make sure they cover as much area as possible. This re-enforces the rule that coverage should be the primary factor when deploying a crowd-sensing platform.

Figure 5.9 is the only case where "good" sensors are placed close to each other: the two green sensors to the north-east. In that case, the two sensors are near a main stage, which had a lot of traffic, but more importantly, each of the sensors covers a different high-traffic street. This means that, the two sensors have a large number of detections that are not redundant.

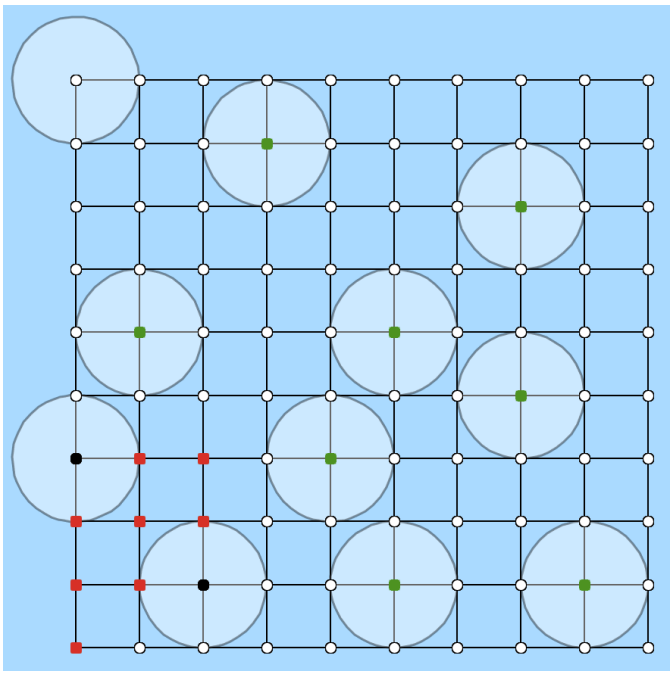


Figure 5.7: Grid map/Simulated (Good sensors green, Bad sensors red, black for both

In all cases the "bad" sensors are packed together, usually in an area of low traffic, such as the corner in the grid map. In the real world, the "bad" sensors are packed in an area with no stages and, as such, with few people and low traffic.

These results reinforce the idea that it is important to spread the sensors while prioritizing high-traffic areas. This ensures the gathering of the highest amount of unique data and information with the smallest number of sensors.

5.6 Summary

In this chapter we studied the effects that the number and position of sensors have on the data gathered using a WiFi remote-positioning system. Data sets are more useful depending on the amount of information that can be extracted from them. We measured how much information can be extracted from a data

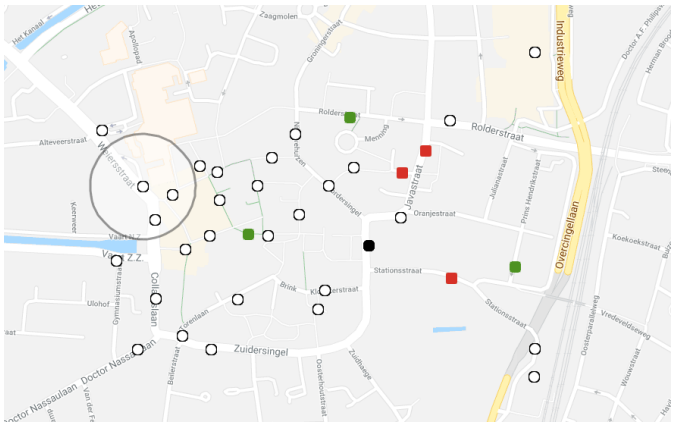


Figure 5.8: Assen map/Simulated (Good sensors green, Bad sensors red, black for both)

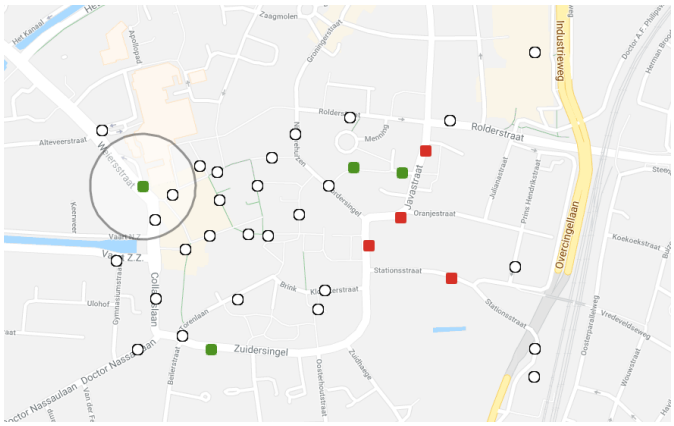


Figure 5.9: Assen map/Real (Good sensors green, Bad sensors red, black for both)

set using the accuracy to correctly label periods of stops and moves.

Our analysis has revealed that using only few sensors, we can extract the same amount of information as we would with many more sensors. **This means that we cannot improve the performance of WiFi remote-positioning systems by adding more sensors.**

We have determined how the number and position of sensors contribute

to obtaining data sets from which we can extract the most crowd-dynamics information. We propose the following scheme for sensor placement:

1. **Determine the high-traffic areas using other methods.** High-traffic areas offer the highest amount of data and are likely to provide most unique detections. Our analysis showed that the number of unique detections is a main factor in increasing the amount of information.

2. **Place one sensor in each high-traffic location, while avoiding overlaps.** Overlaps create redundant data. This makes a sensor placed near an existing one less important than one placed far away.

3. **Place sensors so that they jointly cover as much of the interest area as possible (preferably at least 12-15 sensors per square kilometer - assuming close to 100m detection range).** In all three of the data sets which we analyzed, we reached peak accuracy with as few as 12-15 sensors per square kilometer (see Figure 5.3c). This is achieved under the assumption that the sensors are spread to cover as much of the area as possible.

4. **If a higher number of sensors can be used, consider increasing the interest area or add more to high-traffic areas.** Due to the low probability of recording WiFi frames, having redundant sensors means more unique data can be gathered from the same area. We know the probability of detection is lowered even more in high-traffic areas because our bodies block the WiFi signals.

We believe that these results can easily be transferred to other system that use radio sensors for positioning. Our results and our guidelines **make crowd-dynamics monitoring platforms more accessible by lowering the financial costs of a deployment**. This can be done because we showed most information can be extracted using few sensors.

CHAPTER 6

Sensing Scans versus Connections

Many WiFi remote-positioning implementations record Probe Request frames and nothing else. The assumption is that Probe Request frames are almost always sent. Even when a device is connected to a network, it must send Probe Request frames in order to do roaming and be able to connect to the access points with stronger signals.

We want to test if this assumption is true and if Probe Request frames are enough to represent the total of crowd-dynamics information that can be extracted using WiFi remote positioning. To do this we compare a WiFi remote-positioning data set obtained by recording Probe Requests with one obtained by logging the connection status of devices.

6.1 Contributions

WiFi remote-positioning data sets can be built by recording 802.11 frames. It is common, for the WiFi remote-positioning platforms, to filter frames that are not Probe Request frames. The assumption is that Probe Request frames are always broadcast by mobile devices, even though the frequency might vary. When a device is connected, these frames are broadcast in order to search for better access points. This process of moving from one access point to another is called roaming.

Another way of collecting positioning data from WiFi, while maintaining the same privacy guarantees as with the Probe Request frames, is by using connection logs. Connection logs are kept by most large WiFi network operators (such as Eduroam). **We have performed a data-gathering experiment that collects both Probe Request frames and logs connection status of mobile devices from the same set of sensors**. Because we use the same set of sensors and perform data gathering simultaneously from all of them, the two data sets

describe the same crowd dynamics. We can compare these two data sets to better understand how much information on the crowd dynamics is shared and how much is complementary.

We know that the Probe Request data set contains information on more devices, compared to data sets formed of connection logs, because we cannot expect all devices that move through our sensing area to connect to the sensors' network. After excluding these extra devices from the Probe Request data set, the assumption would be that because the two data sets measure the presence of devices in the same area over the same time, they would offer the same information. If this was the case, and the number of devices detected only through recording Probe Requests would be insignificant, the choice of method would depend only on convenience.

Having the two data sets **we measure to what extent they can be correlated based on time, space and information, in the form of the summary of stops and moves**. We conduct the comparison in order to determine if the assumption that the data sets offer similar information is true.

After determining the validity of the previous assumption, **we explore the possibility of joining the two data sets in order to obtain a more complete picture of crowd dynamics**. We compare the data set obtained after merging with the original ones and present the results in terms of the amount of information that can be extracted.

6.2 Fundamentals

There are many crowd-dynamics monitoring platforms based on WiFi remote-positioning used for commercial deployments or research projects that record only Probe Request frames [66, 68, 72, 79, 90, 133, 134, 135, 136], although WiFi communication uses many other frame types. We presented these frame types and the details of the 802.11 protocol family in Chapter 2.

Recording all frame types is avoided because it raises serious privacy concerns. While Probe Request frames (PR) contain little personal data, such as the MAC of the device and SSIDs of networks it connected to in the past, other frames may contain sensitive data, such as clear text data for unencrypted connections. Due to the privacy concerns in all our data-gathering experiments we limited ourselves to capturing only frame types that do not contain any sensitive data, mostly limiting ourselves to Probe Request frames.

Recording too much meta-data incurs privacy risks due to the possibility of using meta-data to deanonymize the device owners. This is shown in dif-

ferent works [98, 137, 138, 139, 140], where the authors found it is possible to deanonymize devices, even when they use MAC address randomization. In practice, we found that the recorded data is sparse enough to limit such attempts. However, we limit our data gathering to what is absolutely necessary in order to achieve positioning of many anonymous individuals while at the same time ensuring that individuals cannot be identified otherwise.

A privacy-sensitive alternative to capturing Probe Request frames for WiFi remote positioning is to use connection logs. In most large networks, connection logs are recorded to keep track as to when users connect to the network as well as the identifier of the access point they connect to. Having the device identifier, a time stamp and access-point identifier, makes connection log entries equivalent to detections, as we described them in Chapter 2. In order to form a connection, the first step a mobile device needs to take is to start an association, by sending an Association Request frame. Because of this, we call data sets obtained from connection logs, Associations (AS) data sets.

Recording only Probe Request frames is popular because of the assumption that they are transmitted with some regularity. WiFi communication is generally done after a device is connected to a WiFi network, composed of one or multiple access points. Access points broadcast Beacon frames to advertise their availability, and mobile devices can passively scan for the Beacon frames and connect when a Beacon from a known network is received. The 802.11 standard specifies that devices send Probe Request frames in a process called active scanning. When actively scanning, the mobile device searches for access points. This enables connections to networks that do not advertise their availability through Beacon frames and in some cases increases connection speed.

When a device is connected to a network it may continue to actively scan in order to find an access point for the same network but having a stronger signal. Moving from one access point to another of the same network is called roaming. Knowing that active scans can be done even when a device is connected, one can assume that by recording Probe Request frames a device would be detected regardless of its connection state.

The 802.11 standard specifies only that a device can scan, it does not specify when this needs to be done. This decision is left to the operating system and the device manufacturer. Various mobile devices have different rules regarding scanning that depend on the way in which they are used. For instance, laptops are usually turned off or are in standby mode while being transported, and as such cannot actively scan for WiFi networks. Energy saving consideration creates other strict rules of when to actively scan. However, some installed apps that require internet connection can ask the operating system to scan more

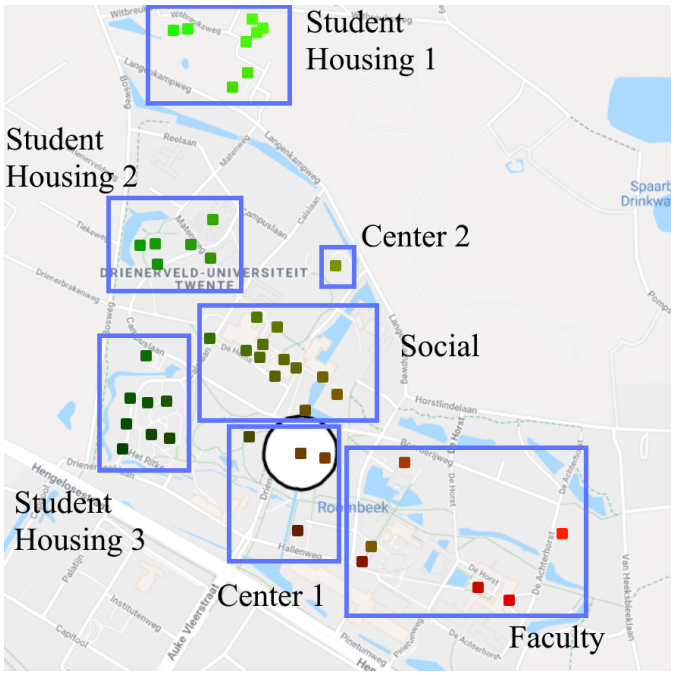


Figure 6.1: Twente Sensor Placement (spots are the sensors, circle is 100m visual guide, squares are sensor groups)

frequently.

We argue that logging connections implies having stronger detections, with a higher positioning accuracy, as devices cannot make and keep connections while near the edge of the area where signals can be correctly received. Building a connection requires complex communication which can only be done if the signals between the mobile device and access point can be correctly received for extended time periods. Furthermore, when communicating, devices need to use their real MAC and send data periodically in order to keep the connection alive. This means that positioning data sets based on connection logs do not contain detections with random MACs and have regular frequencies.

Positioning data sets obtained from Probe Request frames (PR) and those obtained from connection logs, or associations (AS), have different characteristics. PR detections cover more time because they can be recorded even when a

device is not connected but they are highly irregular and offer low positioning accuracy. AS detections are available only when a device is connected but are more regular and have a higher positioning accuracy.

In order to compare PR and AS positioning data sets, we conducted the Twente data-gathering experiment. The two data sets (PR and AS) have positioning data covering the same time period and the same area. The data is gathered by logging connections at a six-minute interval and by sensing and recording Probe Request frames received by the same set of sensors. We gathered WiFi remote-positioning data for three days.

During the Twente data-gathering experiment, the sensors were placed as presented in Figure 6.1. We colored the sensor locations by varying the red color with the latitude and the green with the longitude. Because of this, sensors that are close have similar colors and the ones that are far have different colors. The same color mapping scheme is used later in order to visualize the positioning data. We also grouped the sensors manually, by using our knowledge of the areas (student housing, social and shops, faculty). Each sensor group is surrounded by a square.

6.3 Comparing Probe Requests with Associations

Comparing data sets gathered by recording Probe Request frames (PR) and by using connection logs (AS) is vital. Determining that the two data sets offer different information means that crowd-dynamics measurements using WiFi remote positioning should aim to use both. In contrast, if they offer the same information, we would know that only one of the data sets needs to be gathered. The choice can be made based on convenience. Of course, there is the third possibility, that the data sets are partially similar. In this case we need to determine what percentage of the two data sets is common and what differs.

In order to conduct the comparison, we have identified three criteria based on time, space and a final one, based on information, that combines the two. These choices cover all the properties of the data sets:

- Temporal comparison - **when** are the detections recorded and for **which** devices
- Spatial comparison - **where** are the detections recorded
- Information comparison - **what** information can be extracted, or in other words, what moves, and what stops are described by the positions

6.3.1 Temporal comparison

To make a temporal comparison we take each detection from one of the data sets and try to find a corresponding one in the other data set. For a detection in one data set to correspond to a detection in another, the two detections must be of the same device and match temporally according to a threshold. For now, we ignore the position or the sensor that recorded the detection.

The two data sets have different detection frequencies (6 minutes for AS data set and irregular - from seconds to tens of minutes - for PR data set) and the recording process itself can introduce synchronization differences of a few seconds. Because of this, we decided to say two detections are correspondent if there are less than 3 minutes between them. We chose 3 minutes because it is half the time period between AS detections. If the value was smaller, we would have PR detections between two consecutive AS detections, with no correspondent. If the value was larger, we would have PR detections with multiple AS detection correspondents, possibly creating a chain of corresponding detections. As such, it is guaranteed that each PR detection would have at most one AS detection correspondent. One AS detection can have multiple PR detection correspondents because of the higher frequency.

To summarise, we say a detection λ_i from the detection set of probe requests Λ_{PR} is correspondent with a detection λ_j from the detection set of associations Λ_{AS} if they represent the same device λ^D and are recorded at times λ_T less than 3 minutes from each other. Correspondence is formally described in Equation 6.1.

$$\lambda_i \approx \lambda_j \leftrightarrow \lambda_i \in \Lambda_{PR}; \lambda_j \in \Lambda_{AS}; \lambda_i^D = \lambda_j^D; |\lambda_i^T - \lambda_j^T| < 180s \tag{6.1}$$

To better understand the similarity of detections, we split corresponding detections into three categories: at the same sensor, at sensors that are less than 400m away, and at sensors that are more than 400m away. We chose the distance to be 400m, as it would represent the case where three sensors are placed in a row, each with a sensing radius of 100m. Considering each AS detection can correspond to multiple PR detections, for each AS detection, we select the correspondent that has the smallest geographical distance. Corresponding detections at the same sensor can be considered identical while corresponding detections at more than 400m can be considered invalid. This is because we expect the distance between detections recorded at similar times to be small.

The first column in Figure 6.2a, labeled with Original, represents what percentage of detections from the data sets are correspondent and if they are,

what is the distance between the locations where those detections were recorded. The colors used in the graphs are:

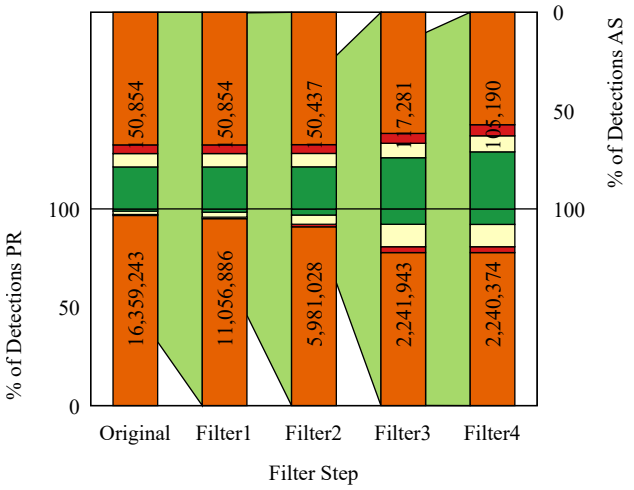
- dark green - perfect match - at least one correspondent detection at the same sensor.
- yellow - partial match - at least one correspondent detection at less than 400m, but no correspondent detection at the same sensor.
- red - mismatch - at least one correspondent detection at more than 400m, and no correspondent detection at less than 400m.
- orange - no correspondent detection.

Consider one detection from the AS data set. If there is no detection in the PR data set of the same device, within three minutes from it, we say it does not have a correspondent and label it with orange. If it does have a correspondent, we find the corresponding detection in the PR data set which is recorded at a sensor closest to the one that recorded the AS detection. Based on the distance, we label the AS detection as correspondent, matching, partially matching or mismatching, with detections from the PR data set. The same is done when considering detections from the PR data set.

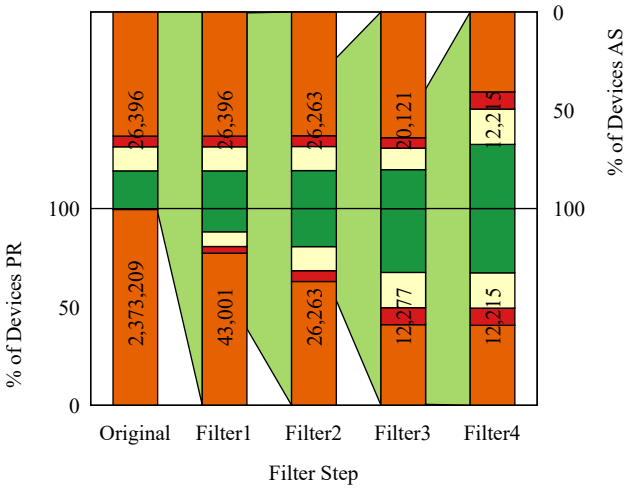
In Figure 6.2b, for the first column, labeled Original, we compare devices by the same criteria. We chose the label color of each device as the one for its best matching detection, in the order from the previous list. We do this because we know most devices have non-correspondent or mismatching detections, based on the percentage presented in Figure 6.2a. This allows us to gain some insight into how many devices have at least part of detections matching or correspondent.

By analyzing the columns labeled *Original* from both Figures 6.2 we observe that the two data sets are extremely different. Most detections do not have correspondents. Furthermore, many devices, especially in the PR data set, do not have even a single corresponding detection.

The data sets consist of 16 million probe request (**PR**) detections and 150 thousand association (**AS**) detections. These detections have over 2 million distinct device identifiers (salted hash MAC addresses) in the PR data set and 26 thousand device identifiers in the AS data set. Detecting 2 million devices is not realistic considering the entire population of Enschede (the city that hosts the Twente University campus) is 150 thousand people and we are recording detections only in the area of the University campus.



(a) Detections



(b) Devices

Figure 6.2: Comparing and filtering the PR and AS data sets. (dark green - match; orange - no correspondent; yellow - partial match; red - mismatch; light green - filtering)

The large differences between the two data sets may be explained by the large number of detections that have attached device identifiers which are constructed given random MAC addresses. Furthermore, some devices may be detected only through one method, or devices might be detected at different times given different methods. We have explained the problems introduced by using random MAC addresses in previous chapters.

For the AS data set, the number of device identifiers is much more realistic. This is because devices need to use a real, unique MAC address when communicating over a WiFi network. In our case, the WiFi network is the one providing WiFi to the University staff, students and guests, called Eduroam.

We know that the number of devices identified through Probe Requests must be larger than the number of devices identified through Associations. This is because everyone who visits the campus and does not connect to the Eduroam network would be present only in the PR data set. Nevertheless, the numbers should be within a reasonable range from each other.

In order to have a more realistic analysis we propose comparing the data sets after filtering out detections that have no chance of having correspondents or to match. For this we apply four filters. Figures 6.2 show how the comparison that we described earlier changes after applying each of the filters. The light green transition shows the effect of each filter. The filters are set in the presented order because after each we gain specific insights about the two data sets. Next, we describe each of the four filters and their effect in detail.

6.3.1.1 Filter 1

Finding the real number of devices detected through Probe Requests is important to understanding the scale that WiFi remote-positioning systems can achieve. We cannot build traces if the devices are using random MAC addresses that are constantly changing. Furthermore, in order to gain a real sense of the difference between the two methods (AS and PR) we need to know the real number of devices detected by each. If it were true that by using PR we could identify 2 million devices compared to 26 thousand in AS, as is apparent in the first column from Figure 6.2b, this would be a strong argument against using AS, because it would offer considerably less information compared to the PR method.

Using standard approaches, such as outlier detection, to remove data recorded with random MAC addresses is not possible because the device identifiers based

on these addresses dominate our data set. IANA [1] has set aside a group of OUIs (first part of the MAC address) that should be used by devices that randomize their MAC address when scanning [141]. This makes it simple to remove detections that belong to random MAC addresses. However, there is no guarantee that only these OUIs are used to form random MAC addresses. Furthermore, due to privacy reasons we do not save the original MAC address, but a salted hash. Without the clear text MAC address, we cannot filter the random ones based on OUI.

By analyzing the data sets, we observed that most of the two million devices have one or few detections. This is consistent with the use of random MAC addresses that change often. If random MAC addresses would not change often, the privacy guarantees that they introduce would not work, as the device could be tracked. The multitude of device identifiers with few detections led us to conclude that because of the large number of available random MAC addresses, for each random MAC address there are only few detections, possibly even just one. This means that it may be possible to find a threshold, so that if there are fewer detections than the threshold for a given device identifier, it is likely to be based on a random MAC address.

We know such a threshold cannot perfectly differentiate between device identifiers based on random MAC addresses and real ones. It is possible for a real device to be detected only once, as such, the threshold would erroneously label its device identifier as one based on a random MAC address. Even worse, due to chance, a random MAC address may be detected more times than a given threshold. Considering no other method is available to differentiate between device identifiers based on real and random MAC addresses, due to privacy considerations, we believe using a threshold offers an acceptable estimate.

The next filter, Filter 2, removes all detections with device identifiers that can be found in only one of the two data sets. This means that it also removes all detections for device identifiers of random MAC addresses. This happens because the AS data set cannot have device identifiers based on random MAC address.

Considering the threshold offers only an estimate, we could ignore Filter 1 and just use Filter 2. However, by applying Filter 1 we discover realistic number of devices that can be detected by either method. Comparing these values is important in order to understand how many more devices can be detected by recording Probe Request frames, even though they never connect to the

[1] IANA: Internet Assigned Numbers Authority `https://www.iana.org/` (Accessed: 01-July-2019)

Eduroam network.

We propose a technique for determining the threshold based on the median number of detections per device. Our thinking is that *comparable data sets (same time frame, same area, same gathering technique) that record real MAC addresses have the same median number of detections per device*. We use median instead of the mean because it is less sensitive to outliers. We know that device identifiers from the PR data set that are also present in the AS data set are based on real MAC addresses. These represent a part of PR device identifiers based on real MAC addresses, but there may potentially be others. Having this information, of a set of device identifiers we know to be based on real MAC addresses, allows us to split the PR data set into a part that has only device identifiers from real MAC addresses and one that has both real and random MAC addresses. We can then filter the one that contains random MAC addresses until we obtain the same median number of detections per device as in the other. The values calculated using the technique are presented in Figure 6.3 and the step by step details are:

- We split the PR data set in two: one would contain detections for all device identifiers that can also be found in the AS data set ($PR^D \in AS^D$); the other would contain the remaining detections ($PR^D \notin AS^D$). We do this because we know that device identifiers based on random MAC addresses can be found only among device identifiers that are not present in the AS data set.
- We calculate the median number of detections per device identifier in the $PR^D \in AS^D$ data subset. This number is used as a reference. For our data set, the median number of detections per device in $PR^D \in AS^D$ was 107 and this is represented using a red line in Figure 6.3.
- We filter detections from the $PR^D \notin AS^D$ data set that have a device identifier for which there are fewer detections than a given threshold. We use all values for the threshold between 0 and 300. We chose not to go beyond 300 as it is a reasonably high number of detections per device identifier based on our experience with the data sets.
- By filtering we obtain 300 data subsets. For each of these data subsets we calculate the median of the number of detections per device identifier. This is shown with blue squares in Figure 6.3.
- We compare the 300 median values with the median number of detections per device calculated based on the filtered $PR^D \in AS^D$ data sets (the 107

value calculated earlier, represented by a red line in Figure 6.3). The threshold used for Filter 1 is the one that makes the median value from $PR^D \notin AS^D$ equal to the one in $PR^D \in AS^D$.

As can be observed in Figure 6.3, marked with a horizontal dashed line, the threshold value obtained through this method is 39. This means that a device identifier for which we have fewer than 39 detections is considered a device identifier based on a random MAC address. All detections we have recorded with the given device identifiers are removed by filter 1. We apply the filter only for device identifiers that do not have detections in the AS data set.

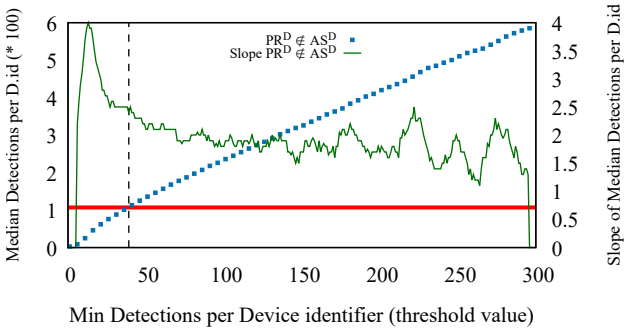


Figure 6.3: Determining threshold for removing data for device identifiers (alias D.id) with few detections
red line - median detections per D.id for PR data subset where the D.id is in AS data set

To confirm our choice of threshold value we used another technique. We calculated the slope of the median of the number of detections per device depending on the threshold (green in Figure 6.3). The knee of the slope is near the threshold value of 39. To the left of the knee the change in median is large, to the right the median values stabilize. This can be explained because most detections, those belonging to random MAC addresses, have already been removed.

Filter 1 removes detections only from the PR data set. To formalize, we use Equation 6.2 to select detections that are to be filtered. Here $|\Lambda_{PR}[\lambda_i^D]|$ is the number of PR detections belonging to device λ_i^D.

$$filteredDetections = \{\lambda_i \in \Lambda_{PR} || \Lambda_{PR}[\lambda_i^D]| < 39\} \tag{6.2}$$

Going back to Figures 6.2, after we apply filter 1 with the threshold set to 39, more than 2 million devices jointly having 5 million detections are removed. This filter affects only non-corresponding detections, the orange part of the PR data set, removing part of them (the start of light green transition).

We are left with 43 thousand device identifiers in the PR data set. This number is much closer to the number of identifiers in the AS data set, 26 thousand. Because the numbers of devices are now reasonably close to each other, we claim that the chosen threshold offers a good estimate.

After applying filter 1 we discovered that the PR data set contains detections for about 65% more devices compared to the AS data set. This is evident in the second column from Figure 6.2. Considering the campus environment and that students and staff have access to the Eduroam network, it is reasonable to assume that about 40-50% of devices do not connect to the network. These devices can belong to guests or be secondary devices for some of the students. Devices that were never configured to connect to the Eduroam network.

What is interesting is that even though a third of the detections were removed, the percentage of detections with corresponding detections in the other data set remains low. This motivates our use of the other three filters.

We estimate that the detections removed by filter 1 represent somewhere around 20 thousand devices. We arrived at this value by dividing the number of filtered detections by the average number of detections per device for the ones that are left. Many of these detections may be of devices that are detected in the AS data set, when using the real MAC address.

6.3.1.2 Filter 2

Filter 2 removes all detections for device identifiers that are found in only one of the two data sets. So far, we discovered that the PR data set contains detections for significantly more devices, but we are interested to learn how the data sets compare for those devices detected by both methods.

To formalize, the detections λ_i that are filtered from the PR data set are represented by the set from Equation 6.3. An equivalent equation represents the set of detections that are removed from the AS data set.

$$filteredDetections = \{\lambda_i \in \Lambda_{PR} | \nexists \lambda_j \in \Lambda_{AS}; \lambda_i^D = \lambda_j^D\} \tag{6.3}$$

When applying filter 2 we discovered that from the 26 thousand devices connected to the Eduroam network only 133 do not have a detection in the PR data set. That amounts to 0.5% of devices in the AS data set. Considering

almost all devices from the AS data set are present in the PR data set, this shows that **most, if not all, devices send Probe Requests with the real MAC, even if they use MAC address randomization**.

We show the distribution of matching detections after applying filter 2 in the 3rd columns from Figures 6.2. The percentages for the AS data set remain almost unchanged, because we removed detections for only 133 devices. However, the PR percentage of matching detections grows noticeably as detections for many devices found only in the PR data set are filtered.

6.3.1.3 Filter 3

The small percentage of matches after applying filter 2 can be explained by devices being detected at different times given the two methods. Consider the following scenario: one device is detected using PR in the morning and AS at night. It is obvious that if we compare the traces of the device, they are different because they represent different time periods. We are interested to perform the comparison on the time period where devices are detected through both methods.

Filter 3 removes all detections that do not belong to the time period in which both methods detect the device. The time periods are calculated for each device independently. We want the time period to start at the latest of the first two detections from the PR and AS traces, and end at the earliest of the last two detections from the PR and AS traces. To formalize, the time period for a device d starts at $startTime_d$ calculated using 6.4 and finishes at $endTime_d$ calculated using 6.5. Detections are filtered using Equation 6.6.

$$startTime_d = max(min(\Lambda_{AS}[d]^T), min(\Lambda_{PR}[d]^T))) \tag{6.4}$$

$$endTime_d = min(max(\Lambda_{AS}[d]^T), max(\Lambda_{PR}[d]^T))) \tag{6.5}$$

$$filteredDetections = \{\lambda_i \in \Lambda[d] | \lambda_i^T < startTime_d \vee endTime_d < \lambda_i^T\} \tag{6.6}$$

To better understand filter 3 we use Figure 6.4. Given the sets of detections (AS and PR) for a device, ordered from left to right by time as we can see in subfigure (a), we select the first and last detection from each data set, subfigure (b). From the four detections we select the latest from the first and earliest from the last, subfigure (c). Finally, we filter all detections that are not between the two timestamps chosen earlier, subfigure (d). It is possible for the two sets of detections to not have anything in common, subfigure (e), or for all detections from one data set to be between all detections from the other, see subfigure (f).

In the last case, our filter would remove all detections for the device from one of the two data sets and leave all in the other.

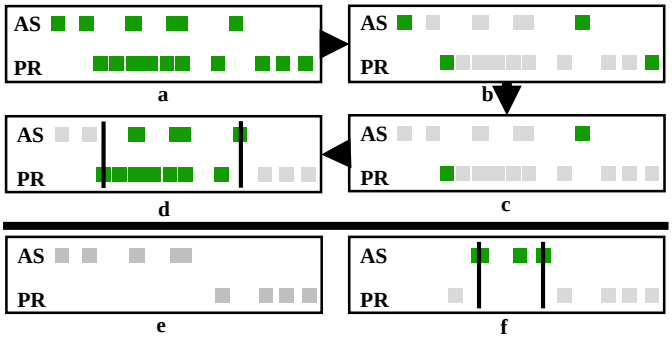


Figure 6.4: Filtering on common time interval

After we apply filter 3 we can observe a noticeable increase in the number of matching detections, in the 4th column from Figure 6.2a. When we analyze the numbers of devices, we observe that, for both data sets, they dropped significantly, from 26 thousand, to 20 thousand in the AS data set and 12 thousand in the PR data set. This shows that for many devices, the two methods gather detections in significantly different time periods.

6.3.1.4 Filter 4

After we apply Filter 3 and remove detections that are not part of the common interval, we are left with different number of devices in the two data sets. Because of this, we apply **Filter 4**. Filter 4 is identical to Filter 2, removing devices that are found in one of the data sets but not the other.

After applying filter 4 we are left with data sets of 12 thousand devices. These 12 thousand devices are represented through 2 million detections in the PR data set as well as 105 thousand detections in the AS data set.

The percentage of detections for which we have no correspondent or for which we have mismatches remains high. This is made evident in the last column from Figure 6.2a. Having a high percentage of detections with no correspondent is surprising, considering that after applying the four filters we removed 30% of detections from AS data set and 86% of detections from the PR data set and all detections removed were non-correspondent.

For devices, in Figure 6.2b, the percentage of them with no correspondent detection is significant. The percentage of non-corresponding or mismatching devices is large, especially considering that we labeled devices based on their best matching detection.

The large amount of data removed by our filters and the differences obtained even after filtering so many detections shows us that the PR and AS data sets are very different, at least when it comes to the recording time of the detections. With such large differences we are inclined to believe the two data sets are more complementary as opposed to similar.

6.3.2 Spatial Comparison

In the previous subsection we compared the two data sets from a temporal perspective. We filtered the original data sets so that we were left with detections in periods where both methods detect the devices. For the spatial comparison, we use the data sets obtained after applying the four filters. The filtered data is not as interesting from a spatial comparison standpoint because it is obvious that if they are detected at completely different times devices can be in different locations.

For the spatial comparison, we take all detections for a device and compare the areas where it is detected. The thinking being that a device should be detected in the same general area by both methods.

We conduct the spatial comparison using two measures, the center of mass of all detections for a device and the radius of gyration. The center of mass $\overrightarrow{R_d}$ for a device d is defined using Equation 6.7, where $\Lambda[d]$ is the set of detections for device d and $\overrightarrow{\lambda^P}$ is a vector representing the geographical position of the sensor (latitude and longitude) at which a detection was recorded. The radius of gyration, $\overrightarrow{G_d}$, as is defined in Equation 6.8, is used in previous works on human movement [142, 143, 1, 144]. The radius of gyration acts like a form of standard deviation of how far the detections are recorded compared to the center of mass. The radius of gyration shows how mobile a device is.

$$\overrightarrow{R_d} = \frac{1}{|\Lambda[d]|} \sum_{\lambda_i \in \Lambda[d]} \overrightarrow{\lambda_i^P} \tag{6.7}$$

$$\overrightarrow{G_d} = \sqrt{\frac{1}{|\Lambda[d]|} \sum_{\lambda_i \in \Lambda[d]} (\overrightarrow{\lambda_i^P} - \overrightarrow{R_d})^2} \tag{6.8}$$

For each device we measure the distance between the two centers of mass (one from each data set). We plotted a histogram of all these distances in Figure 6.5. Many distances are equal to 0, meaning these devices are detected at the same location. However, most devices have positive distances between the centers of mass, reaching even 1km. This means the two data sets show the device in significantly different places. This emphasizes the difference between the two data sets, as not only a temporal one, but also a spatial one.

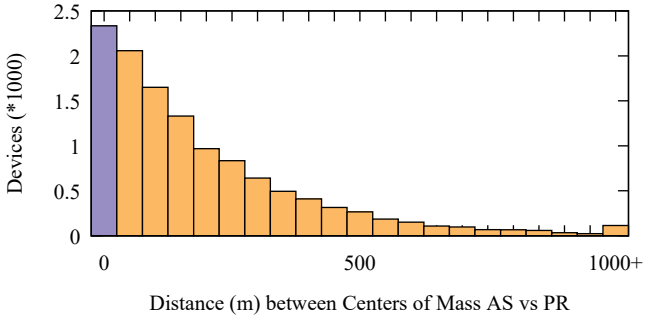


Figure 6.5: Distance between centers of mass (Few centers of mass match - 0m distance)

We compare the radius of gyration for each device (as it results from the two data sets). Figure 6.6 is a 2D histogram showing how many devices have a radius of gyration calculated based on the AS data set (x-axis) and a different one given the PR data set (y-axis). From the figure we can observe that all values are small, meaning the devices have low mobility. We have many cases where devices appear mobile given one data set and static given the other. More so, this characteristic is symmetric, meaning there isn't one data set which generally shows more mobility.

Given the differences between the centers of mass and radius of gyration as they are calculated for each device given the two data sets, we conclude that the two methods (AS and PR) detect the devices in noticeably different areas and with different mobility. This supports the idea that most devices, when connected to a network, no longer scan for other access points in order to do roaming. As far as we know, we are the first to test and conclude this.

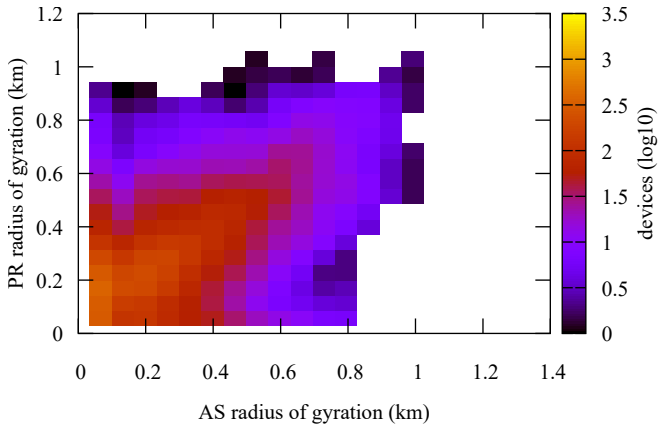


Figure 6.6: Comparing Radius of Gyration

6.3.3 Information Comparison

For our final comparison, we investigate what information can be extracted from detections. Information is best described through the summary of stops and moves. To have a more comprehensive analysis we want to compare not only stops and moves, but also what sensors detect the devices. All the comparisons are presented in Figure 6.7. Next, we go through each and describe the comparison in detail:

- **Dev (Sensors)**. For each device we can extract a set of sensors that detect it. We compare the two sets of sensors (one from the AS data set and one from the PR one). The first column from Figure 6.7 shows the comparison between these sensor sets. Green represents the devices that have all the sensors from one data set present in the other data set. Yellow represents the devices for which only a part of sensors from one data set are present in the other. Finally, orange represents devices for which the sets of sensors that detect them into the two data sets do not have any sensor in common.

  The AS data set has more devices in the green area. This is because the PR data set has more detections at more sensors, making it more likely that some of the sensors that recorded detections of the device in the PR data

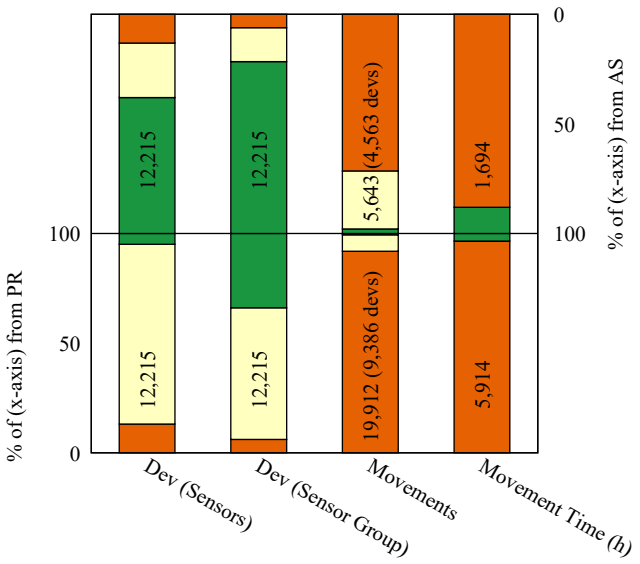


Figure 6.7: Information Comparison

set did not record any in the AS one. We were surprised to see a small, yet significant part of devices that are detected by completely different sensors given the two methods (the orange parts in the AS and PR columns).

- **Dev (Sensor Group)**. It is possible that a lot of differences appear because devices are randomly detected by one of two nearby sensors. To mitigate this we group sensors as we proposed in Figure 6.1. After grouping the sensors, the percentage of devices that are detected by the same sensor groups or by similar ones grows significantly. This can be observed in the second column from Figure 6.7. Yet, there are still some that are detected by completely different sensor groups.

- **Movements**. For the final two columns we analyze stops and moves as was presented in Chapter 4. We set the values of the *Stay Point Detection* algorithm to be: 220m distance between sensors before we consider a move; minStayDuration of 1200, meaning we are not interested in stop periods that are shorter than 20 minutes; and maxMovementDuration of 3600 meaning that for us to consider a movement between two consecutive

detections, the detections must have taken place within an hour. These values were the ones for which we obtained the best results in Chapter 4, where we analyzed stop and move algorithms.

In Figure 6.7 we marked with green all movements of the same device that have similar (within 5 minutes) start and end times in the two data sets. With yellow we mark movements of the same device that start and end in the same sensor group. For the last column, we take all moves from a device within one data set and overlap them with all moves from the other data set. The percentage of matching overlap times are added and marked with dark green. The non-overlapping intervals are marked with orange. A few examples of how move periods can overlap are presented in Figure 6.8.

Figure 6.8: Examples of finding common time for moves
(green - matching; red - not matching)

Because the difference between the two data sets is so large, we conclude that a best practice would be to merge the two data sets in order to obtain a more complete picture.

6.4 Merging the Probe Requests and Associations data sets

Based on the previous comparisons we know that the two data sets are mostly complementary. If we were to merge the two data sets, we would have more detections, over more time, covering more space and offering more information. This implies that in a real deployment one would need to record both Probe Requests and connections to obtain a more complete picture.

We are interested in merging the data sets after the four filters have been applied. The part of the data set presented in the last column of Figure 6.2. The filtered parts from the data sets can be merged, but the results of the merger are obvious and offer no new information. This is because the differences are significant enough so that processing the data set obtained by merging the filtered parts would be identical to processing them separately.

To compare the merged data set with the original ones we count the number of moves obtained by running the stop and move algorithm on the separate data sets (AS/PR) and after running it on the merged data set (AS+PR). The results are presented in the first two columns from Figure 6.9.

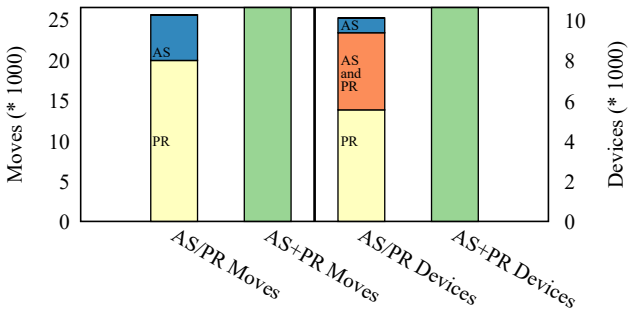


Figure 6.9: Merging PR and AS data sets
blue - moves/devices with moves in AS data set; yellow - moves/devices with moves in the PR data set; green - moves/devices with moves in the merged data set; orange - devices with moves in both the AS and PR data sets

From the merged data set we can extract more moves compared to the sum of moves extracted from the two original data sets separately (see Equation 6.9). Here, $moves()$ represents the stop and move extraction algorithm. The extra moves appear in circumstances where the move starts with a PR detection and ends with an AS one, or the other way around. On the other hand, a small percentage of moves are discovered from the detections generated by both methods. These would in turn lower the number of moves in the merge data set (AS+PR). Equation 6.9 is true given our measurements because the number of matching moves is much smaller than the number of moves added after the merge. There is no guarantee that this is true for all data sets.

$$|moves(AS)| + |moves(PR)| \leq |moves(AS+PR)| \tag{6.9}$$

The last two columns of Figure 6.9 show that the same happens when we look at devices. When we merge the data set, we obtain moves for devices that had no moves in any of the two data sets. For these columns, orange is used to represent devices that have moves in both data sets. Here we can observe the high percentage of devices that have moves in only one of the two data sets, with blue and yellow.

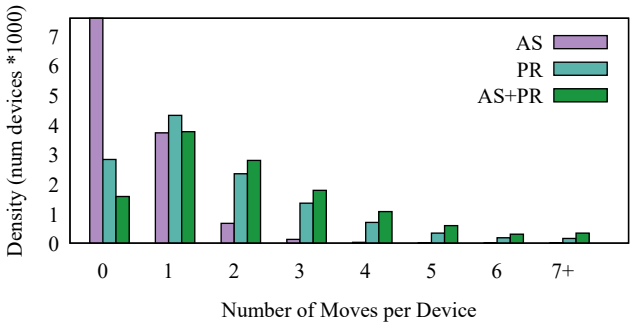


Figure 6.10: Histogram on the number of moves per device

To better understand the effect of merging the data sets before applying the stop and move algorithm we created the histogram shown in Figure 6.10. This histogram shows the number of moves per device depending on the data set used to generate the moves. Here, we can observe that for the AS data set most devices have no moves and those that do, have only one, rarely more. This makes sense considering connections are usually made when we stopped, and we do not visit many distant places during a day.

The merged data set shows more moves per device compared to processing the two data sets separately. However, the number of moves per device remains low. A low number of moves make it difficult to build detailed crowd-dynamics models. The low number can be explained by us having few long moves during the day and more short ones. The short moves are difficult to detect using WiFi remote positioning.

6.5 Explaining the differences

We discovered big differences between the data sets obtained by recording scans (PR) compared to the ones obtained by recording connections (AS). We

expected the PR data set to be larger. Some devices never connect to the network while the ones that do, do not stay connected all the time. When they are not connected, devices can scan for WiFi networks and when they do, they can be detected through the PR method.

The number of detections obtained using the two methods differs widely due to the detection frequency. The frequency of detections for logging associations by our system is 6 minutes. Probe Requests are sent with a higher frequency and at irregular intervals.

What we discovered is that roaming does not function as we initially expected. Considering the 802.11 standard, when a device is connected to a network it can scan for other access points in order to find and transfer to a possibly better connection. Our expectation was that because of the roaming mechanism devices continue to scan while they are connected. During our comparison we discovered this is generally not the case. When a device is connected it does not scan.

Roaming, the main reason for scanning while connected has many challenges [145] and security considerations [146]. Devices avoid scanning because of battery considerations. Take for instance the Android operating system: while connected it scans for alternative access points only when the signal strength for the current connection drops below a threshold. Windows devices let the user configure a "Roaming aggressiveness" setting. Furthermore, some devices have issues with WiFi roaming altogether. This indicates that some, if not most, devices do not scan at all while connected. These issues with roaming for some devices, although known by the mentioned companies and others, as is presented in their websites has not been studied and we have found no scientific paper that addresses it.

Having many devices that do not scan while connected explains why we have so many detections in the AS data set that do not have a matching (similar time, same sensor), or even worse, a correspondent (similar time) detection in the PR data set. This does not completely explain the differences we observed in stops and moves. If a device must scan in order to initially connect to a network, we should see a PR detection before any group of AS detections. Having those PR detections would allow us to detect similar moves. The lack of such detections in the PR data set can be explained by the extensive use of passive scans and the high loss rate of WiFi frames.

When we merge the two data sets and run the stop and moves algorithm on the result, we obtain moves that were not detected when the algorithm was run separately on the two data sets. An example would be the last move of device 3 from Figure 6.11, which we explain next.

We selected five devices to represent in Figure 6.11. For each device we placed a dot at the recording time of a detection in the AS, PR and AS+PR data sets. The color of the dot corresponds to the color of the sensors from the map in Figure 6.1. As we previously explained, a large difference in color translates into a large distance between the sensors that recorded the detections. With blue, we draw lines from the start to the end of a move. For each device we have six rows, from top to bottom: AS detections, AS moves, PR detections, PR moves, AS+PR detections and AS+PR moves. The moves are extracted from the detections drawn on the row above. The black vertical lines represent the interval used for analysis. They were obtained by applying the rules from Figure 6.4.

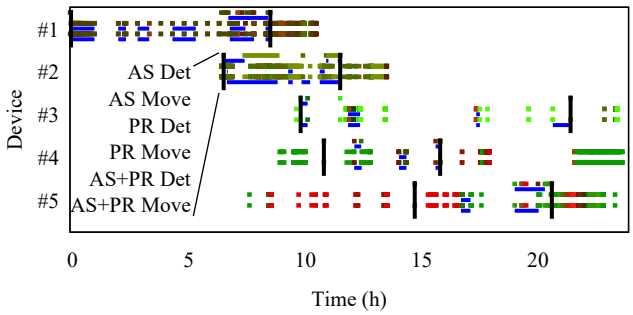


Figure 6.11: Sample of devices detections, movements, for AS, PR and AS+PR (black vars - common time period; blue lines - movement; greens and reds - position of sensor)

In Figure 6.11 we can observe how difficult it is for the algorithm to extract moves. Detections are sparse for all traces. The moves, and the information is extracted based on the chance of having detections at the right time. These six examples are hand-picked to best show different scenarios from our data set, but the sparsity and mismatches are common.

6.6 Summary

We showed that there are large differences between the AS and PR data sets. These differences can be identified at any level, starting from the number of devices detected, number of detections, time and position of detections and even information in the form of stop and moves summary.

We explain what some of the causes for the differences are. Mainly devices

do not send Probe Requests while connected. An interesting discovery was that devices send at least one Probe Request frame with the real MAC address even if they use MAC address randomization.

Given that there are significant differences between the two data sets, even when abstracting the traces to lists of stops and moves, **we explored the possibility of merging the two data sets in order to obtain a more complete picture and have more accurate traces**. When running the stop and move algorithm on the resulting data set, we observed that the list of moves increases beyond the sum of moves extracted originally (from the two data sets separately). This means the traces from the merged data set have finer granularity and more complete information.

Considering most WiFi remote-positioning experiments gather only Probe Request frames, our conclusion, that detections from connection logs add significantly more information, raise important concerns about the completeness of the information in these data sets. Our results show that WiFi remote-positioning data sets gathered by collecting Probe Request frames miss information present in the connection data set along with what we knew is missing information for individuals that cannot be detected at all, either because their device is offline, their WiFi module is stopped or because they do not carry a WiFi-enabled device.

CHAPTER 7

Conclusion and lessons learned

Positioning systems based on existing communication protocols open the way on what can be achieved given large crowd dynamics data sets. They have been used successfully in applications like self-positioning and indoor localization. Furthermore, an extending body of research shows what can be achieved with long-term data extracted by these systems, from facility management and monitoring to security and automatic detection of social groups.

WiFi remote positioning is the most popular of the positioning technologies based on communication protocols. This is because of the average positional accuracy the system offers, offering more precision compared to the low accuracy of GSM/4G systems and higher scalability compared to Bluetooth systems by making it simpler to cover areas. Furthermore, cheap, commercial devices can be repurposed into WiFi remote-positioning sensors, making platform deployments simple and inexpensive.

Even with the high interest in WiFi remote positioning, the characteristics of these systems have not been thoroughly explored. It was obvious that WiFi remote positioning could not be used to monitor every single individual (some people simply do not carry WiFi-enabled devices), but it was not clear how much information can be extracted using WiFi positioning (at least for the people that carry WiFi devices).

In this thesis we showed that given the current assumptions, that there is no control over the target devices and while trying to maintain privacy, the information that can be extracted from crowd-dynamics monitoring systems based on WiFi remote positioning is limited. The limitations are caused by:

A **large number of devices with few detections**. When we analyze the number of detections per device, in all our data sets, we observe that they have a Zipfian-like distribution. This means that few devices have many detections (mostly static), many devices have very few detections and the rest are devices for which we can build traces and extract movement.

The **positional accuracy** is low (equal to the detection range of sensors - about 100m) and techniques such as trilateration that could improve the accuracy can be rarely applied for outdoor environments (less than 10% of detections are simultaneous). Even for simultaneous detections, the variation in RSSI from device to device and based on the environment makes it impossible to increase the accuracy of the resulted position. We showed this in depth in Chapter 3.

Small and varying frequency of recorded positions are the norm considering the target mobile devices are developed to extend battery life and be energy efficient. This leads to huge gaps in traces that cannot be distinguished from the person leaving the detection area.

More **gaps** are introduced when recording only part of the frames received by the sensors (some implementations record only Probe Request frames). These gaps appear because modern devices put an emphasis on energy saving and start scanning for new networks only when the connection quality is low, and broadcasting fewer Probe Request frames.

Low positional accuracy and small frequency of detections leads to **anomalies such as moving in circles**. Missing detections are so frequent in WiFi that it becomes common for devices to be detected at one sensor, then another, then the first again, and so on. We have performed an experiment in our lab with two identical sensors placed 50cm apart and the detections they recorded seemed to show two different worlds, with only a small part of detections matching between the two.

The combination of these properties explains why only a small amount of information can be extracted from WiFi remote-positioning data sets. The amount of tracing information can be correlated with the number of movements which can be extracted. By having to compensate for anomalies like circular movement, we can only extract long movements. In our experience, a movement needs to cover at least 200m for it to be distinguishable from noise (tracing anomalies). This means short walks, such as going across the street, may not be detectable. By analyzing our data sets we observed that traces for most devices do not contain a single movement and those that do, have only few movements.

7.1 Contributions

During our research we addressed the questions from Chapter 1. The answers to the questions are:

Question 1: *Which positioning technology can be used to provide the highest*

amount of data for the highest number of individuals, and, as such, is best suited for monitoring crowd dynamics?

• **Chapter 2:** We conducted a survey of positioning systems. Our survey shows that remote positioning based on repurposed communication technologies is currently the only viable crowd-dynamics monitoring technique because it easily scales to many people. Particularly interesting is WiFi remote-positioning because of the high number of data points it offers compared to the alternatives.

Question 2: *How is WiFi remote positioning implemented and what are the current applications it is used for?*

• **Chapter 2: We conducted a survey on uses for WiFi remote-positioning systems**. We concluded that the technology is popular, especially for indoor applications. Other uses take advantage of data gathered over large time periods. Furthermore, **we described WiFi remote-positioning systems and implementation with more details than previous research**. Interesting details represents the frame types which can be recorded and what channel to use.

Question 3: *What are the properties of traces extracted from data produced by WiFi remote-positioning systems?*

• **Chapter 2: We conducted five data-gathering experiments using WiFi remote positioning**. Using the data from the five data sets, **we showed that positions obtained from WiFi remote-positioning systems are sparse and traces drawn from them have anomalies**. The anomalies give the impression that people are moving in circles. Any attempts to extract information, such as discovering how many people moved from the position of one sensor to the other in a given amount of time are disrupted. This cannot be correctly calculated because the back and forth movement adds up to unrealistic values.

Question 4: *Why are the traces sparse and what are the cyclic-movement anomalies we observe? How can we mitigate the effect caused by said anomalies?*

• **Chapter 3:** ***We showed that the circular-movement anomalies are caused by a combination of low positioning accuracy, and low number of detections.*** This means that a device is not detected by all the sensors that are in its range. It is often that only one of the sensors detects the device, followed by another and so on. The irregular detection range makes it possible for a device to be detected at sensors that are unexpectedly far. ***We showed that traces can be smoothed and most of the circular-movement anomalies can be removed. We compared three algorithms for smoothing traces and compared the results using entropy and dissimilarity to the original trace***.

Question 5: *What useful information can be extracted from positioning data in order to build crowd-dynamics models and how can we quantify this information?*

• **Chapter 4:** We analyzed how traces are summarized in the literature. ***We identified that stops and moves are good representatives of traces***. They represent the summary of a trace and can be controlled by parameters such as minimal stop duration. ***We compared different algorithms for extracting stops and moves and identified "Stay Point Detection" to be the most promising one***. Because they are summaries, the number of stops and moves is a good representative of how much information can be extracted given a crowd-dynamics monitoring platform.

Question 6: *How much crowd-dynamics information can we extract using WiFi remote positioning and how can we increase this value?*

In order to answer question 6, we need to address questions 7 and 8. After we address them, we return to question 6 and draw our final conclusions.

Question 7 (part of question 6): *Can we increase the amount of crowd-dynamics information by adding more sensors and as such, increasing the amount of positioning data?*

• **Chapter 5:** Based on stops and moves we can summarize positioning data sets and find how much information they contain. ***We showed that most available information can be gathered using few sensors***. Increasing the density of sensors does not bring improvements, even worse, in some cases it can add noise which hides movements.

Question 8 (part of question 6): *Can we increase the amount of crowd-dynamics information by using alternative WiFi data sources?*

• **Chapter 6:** ***We compared a data set obtained by gathering probe request frames with one extracted from connection logs***. This showed that both data sets contained much information that was not available in the other one. Furthermore, combining the data sets offered more information, while remaining still underwhelming. Based on these experiments, we discovered that most WiFi remote-positioning deployments are missing significant portions of information that could be extracted if more frame types were recorded or connection logs were used alongside Probe Request frames.

Given the assumption that we cannot modify the communication protocol and we cannot gain access to the target devices, ***we cannot improve the outputs of WiFi remote-positioning systems***.

To readdress question 6: Based on the answers to question 7 and 8 we conclude that WiFi remote positioning offers an underwhelming amount of information for crowd-dynamics systems. We showed that many of the movements are invisible to these systems and there is no room for improvement by adding more sensors as they would only offer redundant data.

Main research question: **To what extent can we model outdoor crowd dynamics based on current positioning technologies?**

Large amounts of positioning data are required in order to build strong crowd-dynamics models. After answering question 1, we have determined WiFi remote positioning to be the technology that offers the most amount of positioning data. We answered question 2 and 3 in order to gain an understanding into WiFi remote-positioning systems. Doing this, we discovered that positioning data provided by these systems is sparse and traces drawn from these data contain anomalies. We explain the anomalies when we answered question 4.

Crowd-dynamics models are stronger if they are based on more information. We answered question 5 in order to find a way to measure how much information can be extracted from positioning data. We discovered that the amount of information is underwhelming and tried to identify ways to increase it. When answering question 6 we discovered that not only can we not increase the amount of information extracted from WiFi remote-positioning systems but the same information can be extracted using fewer sensors (answer to question 7) and a lot of easily accessible information is missing from most positioning data sets (answer to question 8). Even so, the amount of information remains underwhelming.

Based on the findings in each of the chapters: **We conclude that using WiFi remote positioning to gather crowd-dynamics information for short time periods (1 day), for outdoor environments results in models that cannot accurately represent reality. This is mostly due to the sparsity of data and low positioning accuracy given the no-control-of-target nature of the technology.**

7.2 Future Work

Current algorithms for parsing traces formed from positions with low accuracy offer low accuracy. Improving these algorithms is difficult, if not impossible. However, prediction, pattern matching, and interpolation can be used to potentially improve the results. As such, positioning data, even from WiFi, may offer more information.

Applications that require crowd-dynamics information need to be tested against what can be offered through WiFi remote-positioning. For some applications the properties of traces obtained through WiFi technologies may be sufficient. This needs to be determined on a per application basis.

The **difference between outdoor and indoor positioning** using the same WiFi technology should be extensively analyzed. Indoor positioning has the

advantage of working in small areas, meaning two sensors in the same room are likely to detect the same frames. Receiving simultaneous frames enables high accuracy positioning by applying trilateration.

All positioning systems based on communication protocols need to be tested to the same degree. There is a clear potential and interest with regards to new technologies such as 5G. These have the potential of offering more precise positioning due to advancements such as beamforming, which could identify not just the distance from a device, but also the direction.

There is no experiment that tries to **compare the results of positioning technologies based on communication protocols**. One can imagine a data gathering experiment that uses Bluetooth, WiFi and GSM/4G positioning simultaneously. Such an analysis can better describe the capabilities of these systems. Furthermore, we should consider the gains of using multiple such systems simultaneously. Other potential sources may vary. For instance, many social networks gather location data. Positioning data is also stored in most pictures taken with a smartphone. Although the combination of these data sources implies an increase in accuracy and amount of information being gathered, it raises serious privacy concerns that need to be addressed.

As with all monitoring technologies **privacy** remains an open issue. Although our results show that identifying a person using only positioning data from WiFi may be more difficult than expected, the possibility still exists, especially in the case of devices that broadcast at high frequencies. The privacy issues can be explored not just for WiFi, but for all positioning technologies based on repurposed communication protocols.

More large-scale experiments need to be conducted. Especially ones where ground truth data is collected. With a large enough experiment, we can get closer to understanding exactly what percentage of information can be extracted by a positioning technology. Analysis can then reveal if there is a bias in the data gathered introduced by the positioning technology.

New communication protocols, such as 5G need to be considered. As deployment is just starting, we do not currently have enough devices that can use it. This should change in the next few years and with various improvements in the technology, such as beamforming, it will become extremely attractive for monitoring crowds. The use of 5G may imply a larger number of devices that can be tracked as well as increased positioning accuracy.

Bibliography

[1] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *nature*, vol. 453, no. 7196, p. 779, 2008.

[2] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global positioning system: theory and practice*. Springer Science & Business Media, 2012.

[3] K.-T. Chang, *Introduction to geographic information systems*. McGraw-Hill Higher Education Boston, 2006.

[4] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1. Ieee, 2004, pp. I–I.

[5] F. H. Raab, E. B. Blood, T. O. Steiner, and H. R. Jones, "Magnetic position and orientation tracking system," *IEEE Transactions on Aerospace and Electronic systems*, no. 5, pp. 709–718, 1979.

[6] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1281–1293, 2013.

[7] B. W. Parkinson, P. Enge, P. Axelrad, and J. J. Spilker Jr, *Global positioning system: Theory and applications, Volume II*. American Institute of Aeronautics and Astronautics, 1996.

[8] K. Vickery, "Acoustic positioning systems. a practical overview of current systems," in *Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles (Cat. No. 98CH36290)*. IEEE, 1998, pp. 5–17.

[9] M. Cablk, J. Sagebiel, J. Heaton, and C. Valentin, "Olfaction-based detection distance: a quantitative analysis of how far away dogs recognize tortoise odor and follow it to source," *Sensors*, vol. 8, no. 4, pp. 2208–2222, 2008.

[10] J. Hallberg, M. Nilsson, and K. Synnes, "Positioning with bluetooth," in *International Conference on Telecommunications: Special Session on IoT Emerging Technologies: Design and Security (ITEMS'16) 16/05/2016-18/05/2016*. IEEE Communications Society, 2003, pp. 954–958.

[11] C. Yang and H.-R. Shao, "Wifi-based indoor positioning," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, 2015.

[12] C. Drane, M. Macnaughtan, and C. Scott, "Positioning gsm telephones," *IEEE Communications magazine*, vol. 36, no. 4, pp. 46–54, 1998.

[13] C. L. F. Mayorga, F. Della Rosa, S. A. Wardana, G. Simone, M. C. N. Raynal, J. Figueiras, and S. Frattasi, "Cooperative positioning techniques for mobile localization in 4g cellular networks," in *IEEE international conference on pervasive services*. IEEE, 2007, pp. 39–44.

[14] "Ieee standard for information technology–telecommunications and information exchange between systems local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1–3534, Dec 2016.

[15] P. A. Zandbergen, "Accuracy of iphone locations: A comparison of assisted gps, wifi and cellular positioning," *Transactions in GIS*, vol. 13, pp. 5–25, 2009.

[16] C. Hurley, *Wardriving: Drive, detect, defend: A guide to wireless security*. Elsevier, 2004.

[17] Google, *Android using WiFi for localization*, 2019 (accessed April 3, 2019), https://developer.android.com/guide/topics/location/strategies.

[18] Apple, *iOS using WiFi for localization*, 2019 (accessed April 3, 2019), https://support.apple.com/en-us/HT203033.

[19] E. Schenk, C. Guittard *et al.*, "Crowdsourcing: What can be outsourced to the crowd, and why," in *Workshop on open source innovation, Strasbourg, France*, vol. 72. Citeseer, 2009, p. 3.

[20] M. B. Kjærgaard, M. Wirz, D. Roggen, and G. Tröster, "Mobile sensing of pedestrian flocks in indoor environments using wifi signals," in *2012 IEEE International Conference on Pervasive Computing and Communications*. IEEE, 2012, pp. 95–102.

[21] P. Sapiezynski, A. Stopczynski, R. Gatej, and S. Lehmann, "Tracking human mobility using wifi signals," *PloS one*, vol. 10, no. 7, p. e0130824, 2015.

[22] J. Rekimoto, T. Miyaki, and T. Ishizawa, "Lifetag: Wifi-based continuous location logging for life pattern analysis," in *LoCA*, vol. 2007, 2007, pp. 35–49.

[23] V. Sekara and S. Lehmann, "The strength of friendship ties in proximity sensor data," *PloS one*, vol. 9, no. 7, p. e100915, 2014.

[24] T. S. Prentow, A. J. Ruiz-Ruiz, H. Blunck, A. Stisen, and M. B. Kjærgaard, "Spatio-temporal facility utilization analysis from exhaustive wifi monitoring," *Pervasive and Mobile Computing*, vol. 16, pp. 305–316, 2015.

[25] N. Husted and S. Myers, "Mobile location tracking in metro areas: malnets and others," in *Proceedings of the 17th ACM conference on Computer and communications security*. ACM, 2010, pp. 85–96.

[26] B. Davies and R. Harré, "Positioning: The discursive production of selves," *Journal for the theory of social behaviour*, vol. 20, no. 1, pp. 43–63, 1990.

[27] D. W. Cravens and N. Piercy, *Strategic marketing*. McGraw-Hill New York, 2006, vol. 6.

[28] D. M. Eigler and E. K. Schweizer, "Positioning single atoms with a scanning tunnelling microscope," *Nature*, vol. 344, no. 6266, p. 524, 1990.

[29] C. F. Serago, S. J. Chungbin, S. J. Buskirk, G. A. Ezzell, A. C. Collie, and S. A. Vora, "Initial experience with ultrasound localization for positioning prostate cancer patients for external beam radiotherapy," *International Journal of Radiation Oncology* Biology* Physics*, vol. 53, no. 5, pp. 1130–1138, 2002.

[30] L. A. Parada, P. G. McQueen, P. J. Munson, and T. Misteli, "Conservation of relative chromosome positioning in normal and cancer cells," *Current Biology*, vol. 12, no. 19, pp. 1692–1697, 2002.

[31] A. Mazzoldi, D. De Rossi, F. Lorussi, E. Scilingo, and R. Paradiso, "Smart textiles for wearable motion capture systems," *AUTEX Research Journal*, vol. 2, no. 4, pp. 199–203, 2002.

[32] S. L. Delp, J. P. Loan, C. B. Robinson, A. Y. Wong, and S. D. Stulberg, "Computer-assisted surgical system," Nov. 4 1997, uS Patent 5,682,886.

[33] J. C. Lee, "Hacking the nintendo wii remote," *IEEE pervasive computing*, vol. 7, no. 3, pp. 39–45, 2008.

[34] Z. Zhang, "Microsoft kinect sensor and its effect," *IEEE multimedia*, vol. 19, no. 2, pp. 4–10, 2012.

[35] M. McCahill and C. Norris, "Cctv in london," *Report deliverable of UrbanEye project*, 2002.

[36] N. T. Siebel and S. Maybank, "The advisor visual surveillance system," in *ECCV 2004 workshop applications of computer vision (ACV)*, vol. 1. Citeseer, 2004.

[37] R. Armitage, "To cctv or not to cctv," *A review of current research into the effectiveness of CCTV systems in reducing crime*, p. 8, 2002.

[38] R. Mautz and S. Tilch, "Survey of optical indoor positioning systems," in *2011 international conference on indoor positioning and indoor navigation*. IEEE, 2011, pp. 1–7.

[39] A. Chaaraoui, J. Padilla-López, F. Ferrández-Pastor, M. Nieto-Hidalgo, and F. Flórez-Revuelta, "A vision-based system for intelligent monitoring: human behaviour analysis and privacy by context," *Sensors*, vol. 14, no. 5, pp. 8895–8925, 2014.

[40] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings. 1985 IEEE international conference on robotics and automation*, vol. 2. IEEE, 1985, pp. 116–121.

[41] M. I. Skolnik, "Introduction to radar," *Radar handbook*, vol. 2, p. 21, 1962.

[42] R. Belderson and A. Stride, "The shape of submarine canyon heads revealed by asdic," in *Deep Sea Research and Oceanographic Abstracts*, vol. 16, no. 1. Elsevier, 1969, pp. 103–104.

[43] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, 1983.

[44] S. S. Blackman, "Multiple-target tracking with radar applications," *Dedham, MA, Artech House, Inc., 1986, 463 p.*, 1986.

[45] A. Ward, A. Jones, and A. Hopper, "A new location technique for the active office," *IEEE Personal communications*, vol. 4, no. 5, pp. 42–47, 1997.

[46] W. Xi, J. Zhao, X.-Y. Li, K. Zhao, S. Tang, X. Liu, and Z. Jiang, "Electronic frog eye: Counting crowd using wifi," in *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2014, pp. 361–369.

[47] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS–global navigation satellite systems: GPS, GLONASS, Galileo, and more*. Springer Science & Business Media, 2007.

[48] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems (TOIS)*, vol. 10, no. 1, pp. 91–102, 1992.

[49] C. Martella, A. Miraglia, M. Cattani, and M. van Steen, "Leveraging Proximity Sensing to Mine the Behavior of Museum Visitors," in *IEEE International Conference on Pervasive Computing and Communication*. IEEE Computer Society, Mar. 2016.

[50] M. Mouly, M.-B. Pautet, and T. Foreword By-Haug, *The GSM system for mobile communications*. Telecom publishing, 1992.

[51] O. Järv, R. Ahas, and F. Witlox, "Understanding monthly variability in human activity spaces: A twelve-month study using mobile phone call detail records," *Transportation Research Part C: Emerging Technologies*, vol. 38, pp. 122–135, 2014.

[52] C. Smith, *3G wireless networks*. McGraw-Hill, Inc., 2006.

[53] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-advanced for mobile broadband*. Academic press, 2013.

[54] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. Soong, and J. C. Zhang, "What will 5g be?" *IEEE Journal on selected areas in communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[55] B. P. Crow, I. Widjaja, J. G. Kim, and P. T. Sakai, "Ieee 802.11 wireless local area networks," *IEEE Communications magazine*, vol. 35, no. 9, pp. 116–126, 1997.

[56] J. C. Haartsen, "Bluetooth radio system," *Wiley Encyclopedia of Telecommunications*, 2003.

[57] Y. Wang, X. Yang, Y. Zhao, Y. Liu, and L. Cuthbert, "Bluetooth positioning using rssi and triangulation methods," in *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*. IEEE, 2013, pp. 837–842.

[58] L. L. Peterson and B. S. Davie, *Computer networks: a systems approach*. Elsevier, 2007.

[59] P. Arana, "Benefits and vulnerabilities of wi-fi protected access 2 (wpa2)," *INFS*, vol. 612, pp. 1–6, 2006.

[60] Z. Cao, Q. Ma, A. B. Smolders, Y. Jiao, M. J. Wale, C. W. Oh, H. Wu, and A. M. J. Koonen, "Advanced integration techniques on broadband millimeter-wave beam steering for 5g wireless networks and beyond," *IEEE journal of quantum electronics*, vol. 52, no. 1, pp. 1–20, 2015.

[61] D. Mills, J. Martin, J. Burbank, and W. Kasch, "Rfc 5905: Network time protocol version 4: Protocol and algorithms specification. internet engineering task force (ietf), 2010," *tools. ietf. org/html/rfc5905*.

[62] R. Rivest, "The md5 message-digest algorithm," 1992.

[63] M. Cunche, "I know your mac address: Targeted tracking of individual using wi-fi," *Journal of Computer Virology and Hacking Techniques*, vol. 10, no. 4, pp. 219–227, 2014.

[64] C. Matte and M. Cunche, "Wombat: An experimental wi-fi tracking system," 2017.

[65] B. Bonne, A. Barzan, P. Quax, and W. Lamotte, "Wifipi: Involuntary tracking of visitors at mass events," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*. IEEE, 2013, pp. 1–6.

[66] J. Scheuner, G. Mazlami, D. Schöni, S. Stephan, A. De Carli, T. Bocek, and B. Stiller, "Probr-a generic and passive wifi tracking system," in *2016 IEEE 41st Conference on Local Computer Networks (LCN)*. IEEE, 2016, pp. 495–502.

[67] F. Hong, Y. Zhang, Z. Zhang, M. Wei, Y. Feng, and Z. Guo, "Wap: Indoor localization and tracking using wifi-assisted particle filter," in *Local Computer Networks (LCN), 2014 IEEE 39th Conference on*. IEEE, 2014, pp. 210–217.

[68] K. Li, C. Yuen, S. S. Kanhere, K. Hu, W. Zhang, F. Jiang, and X. Liu, "An experimental study for tracking crowd in smart cities," *IEEE Systems Journal*, 2018.

[69] E. Furey, K. Curran, and P. McKevitt, "Habits: a bayesian filter approach to indoor tracking and location," in *Proc. of the 22nd Irish Conference on Artificial Intelligence and Cognitive Science (AICS-2011)*, 2011, pp. 11–25.

[70] H. Zou, Y. Zhou, J. Yang, W. Gu, L. Xie, and C. Spanos, "Freecount: Device-free crowd counting with commodity wifi," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[71] A.-C. Petre, C. Chilipirea, M. Baratchi, C. Dobre, and M. van Steen, "Wifi tracking of pedestrian behavior," in *Smart Sensors Networks*. Elsevier, 2017, pp. 309–337.

[72] A. E. Redondi and M. Cesana, "Building up knowledge through passive wifi probes," *Computer Communications*, vol. 117, pp. 1–12, 2018.

[73] F. Potortì, A. Crivello, M. Girolami, E. Traficante, and P. Barsocchi, "Wi-fi probes as digital crumbs for crowd localisation," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2016, pp. 1–8.

[74] J. Weppner, B. Bischke, and P. Lukowicz, "Monitoring crowd condition in public spaces by tracking mobile consumer devices with wifi interface," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*. ACM, 2016, pp. 1363–1371.

[75] X. Tang, B. Xiao, and K. Li, "Indoor crowd density estimation through mobile smartphone wi-fi probes," *IEEE transactions on systems, man, and cybernetics: systems*, 2018.

[76] W. Wang, J. Chen, T. Hong, and N. Zhu, "Occupancy prediction through markov based feedback recurrent neural network (m-frnn) algorithm with wifi probe technology," *Building and Environment*, vol. 138, pp. 160–170, 2018.

[77] T. Kulshrestha, D. Saxena, R. Niyogi, and J. Cao, "Real-time crowd monitoring using seamless indoor-outdoor localization," *IEEE Transactions on Mobile Computing*, 2019.

[78] S. Depatla and Y. Mostofi, "Crowd counting through walls using wifi," in *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2018, pp. 1–10.

[79] V. Acuna, A. Kumbhar, E. Vattapparamban, F. Rajabli, and I. Guvenc, "Localization of wifi devices using probe requests captured at unmanned aerial vehicles," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2017, pp. 1–6.

[80] A. J. Ruiz-Ruiz, H. Blunck, T. S. Prentow, A. Stisen, and M. B. Kjærgaard, "Analysis methods for extracting knowledge from large-scale wifi monitoring to inform building facility planning," in *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*. IEEE, 2014, pp. 130–138.

[81] N. Nunes, M. Ribeiro, C. Prandi, and V. Nisi, "Beanstalk: a community based passive wi-fi tracking system for analysing tourism dynamics," in *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. ACM, 2017, pp. 93–98.

[82] A. Musa and J. Eriksson, "Tracking unmodified smartphones using wi-fi monitors," in *Proceedings of the 10th ACM conference on embedded network sensor systems*. ACM, 2012, pp. 281–294.

[83] T. A. Myrvoll, J. E. Håkegård, T. Matsui, and F. Septier, "Counting public transport passenger using wifi signatures of mobile devices," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–6.

[84] L. Mikkelsen, R. Buchakchiev, T. Madsen, and H. P. Schwefel, "Public transport occupancy estimation using wlan probing," in *2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM)*. IEEE, 2016, pp. 302–308.

[85] W. Pattanusorn, I. Nilkhamhang, S. Kittipiyakul, K. Ekkachai, and A. Takahashi, "Passenger estimation system using wi-fi probe request," in *2016 7th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*. IEEE, 2016, pp. 67–72.

[86] P. Reichl, B. Oh, R. Ravitharan, and M. Stafford, "Using wifi technologies to count passengers in real-time around rail infrastructure," in *2018 International Conference on Intelligent Rail Transportation (ICIRT)*. IEEE, 2018, pp. 1–5.

[87] G. Vanderhulst, A. Mashhadi, M. Dashti, and F. Kawsar, "Detecting human encounters from wifi radio signals," in *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*. ACM, 2015, pp. 97–108.

[88] H. Hong, C. Luo, and M. C. Chan, "Socialprobe: Understanding social interaction through passive wifi monitoring," in *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ACM, 2016, pp. 94–103.

[89] M. Cunche, M. A. Kaafar, and R. Boreli, "I know who you will meet this evening! linking wireless devices using wi-fi probe requests," in *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2012, pp. 1–9.

[90] M. Cunche, M.-A. Kaafar, and R. Boreli, "Linking wireless devices using information contained in wi-fi probe requests," *Pervasive and Mobile Computing*, vol. 11, pp. 56–69, 2014.

[91] L. Schauer and C. Linnhoff-Popien, "Extracting context information from wi-fi captures," in *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, 2017, pp. 123–130.

[92] A. E. C. Redondi, D. Sanvito, and M. Cesana, "Passive classification of wi-fi enabled devices," in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. ACM, 2016, pp. 51–58.

[93] A. Mashhadi, G. Vanderhulst, U. G. Acer, and F. Kawsar, "An autonomous reputation framework for physical locations based on wifi signals," in *Proceedings of the 2nd workshop on Workshop on Physical Analytics*. ACM, 2015, pp. 43–46.

[94] M. Baratchi, G. Heijenk, and M. van Steen, "Spaceprint: a mobility-based fingerprinting scheme for public spaces," *arXiv preprint arXiv:1703.09962*, 2017.

[95] M. Luca and G. Zervas, "Fake it till you make it: Reputation, competition, and yelp review fraud," *Management Science*, vol. 62, no. 12, pp. 3412–3427, 2016.

[96] P. Prasertsung and T. Horanont, "How does coffee shop get crowded?: Using wifi footprints to deliver insights into the success of promotion," in *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM, 2017, pp. 421–426.

[97] M. Vanhoef, C. Matte, M. Cunche, L. S. Cardoso, and F. Piessens, "Why mac address randomization is not enough: An analysis of wi-fi network discovery mechanisms," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM, 2016, pp. 413–424.

[98] A. Di Luzio, A. Mei, and J. Stefa, "Mind your probes: De-anonymization of large crowds through smartphone wifi probe requests," in *The 35th Annual IEEE International Conference on Computer Communications INFOCOM*, 2016, pp. 1–9.

[99] C. Chen, Y. Chen, Y. Han, H.-Q. Lai, and K. R. Liu, "Achieving centimeter-accuracy indoor localization on wifi platforms: A frequency hopping approach," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 111–121, 2016.

[100] Z. Li, T. Braun, and D. C. Dimitrova, "A passive wifi source localization system based on fine-grained power-based trilateration," in *2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2015, pp. 1–9.

[101] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spoton: Indoor localization using commercial off-the-shelf wifi nics," in *IPSN*, 2015.

[102] J. Xu, W. Liu, F. Lang, Y. Zhang, and C. Wang, "Distance measurement model based on rssi in wsn," *Wireless Sensor Network*, vol. 2, no. 08, p. 606, 2010.

[103] E. Mok and G. Retscher, "Location determination using wifi fingerprinting versus wifi trilateration," *Journal of Location Based Services*, vol. 1, no. 2, pp. 145–159, 2007.

[104] N. Le Dortz, F. Gain, and P. Zetterberg, "Wifi fingerprint indoor positioning system using probability distribution comparison," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2301–2304.

[105] Y. Kim, H. Shin, and H. Cha, "Smartphone-based wi-fi pedestrian-tracking system tolerating the rss variance problem," in *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*. IEEE, 2012, pp. 11–19.

[106] A. Bose and C. H. Foh, "A practical path loss model for indoor wifi positioning enhancement," in *2007 6th International Conference on Information, Communications & Signal Processing*. IEEE, 2007, pp. 1–5.

[107] R. K. Sheshadri and D. Koutsonikolas, "On packet loss rates in modern 802.11 networks," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[108] D. Murray, T. Koziniec, M. Dixon, and K. Lee, "Measuring the reliability of 802.11 wifi networks," in *2015 Internet Technologies and Applications (ITA)*. IEEE, 2015, pp. 233–238.

[109] X. Hu, L. Song, D. Van Bruggen, and A. Striegel, "Is there wifi yet?: How aggressive probe requests deteriorate energy and throughput," in *Proceedings of the 2015 Internet Measurement Conference*. ACM, 2015, pp. 317–323.

[110] D. Jaisinghani, V. Naik, S. K. Kaul, and S. Roy, "Realtime detection of degradation in wifi network's goodput due to probe traffic," in *2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE, 2015, pp. 42–47.

[111] M. Boukhechba, A. Bouzouane, B. Bouchard, C. Gouin-Vallerand, and S. Giroux, "Online recognition of people's activities from raw gps data: Semantic trajectory data analysis," in *Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments*. ACM, 2015, p. 40.

[112] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo, and J. Eriksson, "Vtrack: accurate, energy-aware road traffic delay estimation using mobile phones," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2009, pp. 85–98.

[113] Z. Yan, C. Parent, S. Spaccapietra, and D. Chakraborty, "A hybrid model and computing platform for spatio-semantic trajectories," in *The Semantic Web: Research and Applications*. Springer, 2010, pp. 60–75.

[114] L. Schauer, M. Werner, and P. Marcus, "Estimating crowd densities and pedestrian flows using wi-fi and bluetooth," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014, pp. 171–177.

[115] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[116] A. T. Palma, V. Bogorny, B. Kuijpers, and L. O. Alvares, "A clustering-based approach for discovering interesting places in trajectories," in *Proceedings of the ACM symposium on Applied computing*, 2008, pp. 863–868.

[117] J. A. M. R. Rocha, V. C. Times, G. Oliveira, L. O. Alvares, and V. Bogorny, "DB-SMoT: A direction-based spatio-temporal clustering method," *2010 IEEE International Conference on Intelligent Systems, IS 2010 - Proceedings*, pp. 114–119, 2010.

[118] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello, "Extracting places from traces of locations," in *Proceedings of the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots*. ACM, 2004, pp. 110–118.

[119] M. Modsching, R. Kramer, and K. ten Hagen, "Field trial on gps accuracy in a medium size city: The influence of built-up," in *3rd workshop on positioning, navigation and communication*, vol. 2006, 2006, pp. 209–218.

[120] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[121] Q. Li, Y. Zheng, X. Xie, Y. Chen, W. Liu, and W. Ma, "Mining user similarity based on location history," *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, no. c, p. 34, 2008.

[122] D. D. Harrison and G. W. Brooksby, "Method and apparatus for locating an object using reduced number of gps satellite signals or with improved accuracy," Nov. 2 1999, uS Patent 5,977,909.

[123] I. A. Hulbert and J. French, "The accuracy of gps for wildlife telemetry and habitat mapping," *Journal of Applied Ecology*, vol. 38, no. 4, pp. 869–878, 2001.

[124] P. Sigrist, P. Coppin, and M. Hermy, "Impact of forest canopy on quality and accuracy of gps measurements," *International Journal of Remote Sensing*, vol. 20, no. 18, pp. 3595–3610, 1999.

[125] H. Park and A. Haghani, "Optimal number and location of Bluetooth sensors considering stochastic travel time prediction," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 203–216, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.trc.2015.03.023

[126] G. Capellari, E. Chatzi, S. Mariani, P. Milano, I. Civile, and P. L. Vinci, "Optimal Sensor Placement through Bayesian Experimental Design : Effect of Measurement Noise and Number of Sensors †," pp. 2–7, 2017.

[127] J. Sun and R. Wang, "Sensors Layout Optimization Based on Cluster Analysis in Water Supply Network," no. Iceta, pp. 1011–1015, 2016.

[128] "Sensor placement for effective coverage and surveillance in distributed sensor networks," *IEEE Wireless Communications and Networking Conference, WCNC*, vol. 3, no. C, pp. 1609–1614, 2003.

[129] L. Kong, M. Zhao, X.-y. Liu, and J. Lu, "Surface Coverage in Wireless Sensor Networks," *Infocom 2009, Ieee*, vol. 25, no. 1, pp. 234–243, 2009.

[130] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in wireless Ad-Hoc sensor networks," *Proceedings of the 7th annual international conference on Mobile computing and networking - MobiCom '01*, pp. 139–150, 2001. [Online]. Available: http://portal.acm.org/citation.cfm?doid=381677.381691

[131] V. I. Nistorica, C. Chilipirea, and C. Dobre, "How many people are needed for a crowdsensing campaign?" in *2016 IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, 2016, pp. 353–358.

[132] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[133] E. Vattapparamban, B. S. Çiftler, I. Güvenç, K. Akkaya, and A. Kadri, "Indoor occupancy tracking in smart buildings using passive sniffing of probe requests," in *2016 IEEE International Conference on Communications Workshops (ICC)*. IEEE, 2016, pp. 38–44.

[134] A. Alessandrini, C. Gioia, F. Sermi, I. Sofos, D. Tarchi, and M. Vespe, "Wifi positioning and big data to monitor flows of people on a wide scale," in *2017 European Navigation Conference (ENC)*. IEEE, 2017, pp. 322–328.

[135] L. Sun, S. Chen, Z. Zheng, and L. Xu, "Mobile device passive localization based on ieee 802.11 probe request frames," *Mobile Information Systems*, vol. 2017, 2017.

[136] H. Chen, Y. Zhang, W. Li, and P. Zhang, "Non-cooperative wi-fi localization via monitoring probe request frames," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2016, pp. 1–5.

[137] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown, "A study of mac address randomization in mobile devices and when it fails," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 365–383, 2017.

[138] P. Robyns, B. Bonné, P. Quax, and W. Lamotte, "Noncooperative 802.11 mac layer fingerprinting and tracking of mobile devices," *Security and Communication Networks*, vol. 2017, 2017.

[139] C. Matte and M. Cunche, "Panoptiphone: How unique is your wi-fi device?" in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2016, pp. 209–211.

[140] J. Freudiger, "How talkative is your mobile device?: an experimental study of wi-fi probe requests," in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2015, p. 8.

[141] IEEE. (2017) Guidelines for use of extended unique identifier(eui), organizationally unique identifier (oui), and company id(cid). [Online]. Available: https://standards.ieee.org/content/dam/ieee-standards/standards/web/documents/tutorials/eui.pdf

[142] W. Sun, D. Miao, X. Qin, and G. Wei, "Characterizing user mobility from the view of 4g cellular network," in *2016 17th IEEE International Conference on Mobile Data Management (MDM)*, vol. 1. IEEE, 2016, pp. 34–39.

[143] G. Ranjan, H. Zang, Z.-L. Zhang, and J. Bolot, "Are call detail records biased for sampling human mobility?" *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 16, no. 3, pp. 33–44, 2012.

[144] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[145] S. J. Vaughan-Nichols, "The challenge of wi-fi roaming," *Computer*, vol. 36, no. 7, pp. 17–19, 2003.

[146] R. Robert, M. Manulis, F. De Villenfagne, D. Leroy, J. Jost, F. Koeune, C. Ker, J.-M. Dinant, Y. Poullet, O. Bonaventure *et al.*, "Wifi roaming: Legal implications and security constraints," *International Journal of Law and Information Technology*, vol. 16, no. 3, pp. 205–241, 2008.

About the author

Cristian Chilipirea obtained his B.Sc. degree in Computer Science from the Politehnica University of Bucharest, Romania, in 2012. Later, he obtained his M.Sc. degrees in Parallel and Distributed Computing from University Politehnica of Bucharest, Romania and Vrije University of Amsterdam, Netherlands. In 2014, he started the pursue of his Ph.D. degree crowd analysis at University Politehnica of Bucharest, Romania and University of Twente, Enschede, Netherlands.

List of publications in which he participated in reverse chronological order:

1) ***Journal 3.031 IF - Q1***: **Cristian Chilipirea**, Mitra Baratchi, Ciprian Dobre, and Maarten Van Steen, "Identifying stops and moves in wifi tracking data", Sensors, vol. 18, no. 11, p. 4039, 2018, **(Chapter 4 is based on this work)**

2) **Cristian Chilipirea**, Mitra Baratchi, Ciprian Dobre, and Maarten Van Steen, "Identifying movements in noisy crowd analytics data", 19th IEEE International Conference on Mobile Data Management (MDM), p. 161-166, 2018, **(Chapter 4 is based on this work)**

3) ***Journal 1.045 IF - Q3***: **Cristian Chilipirea**, Andreea-Cristina Petre, Loredana-Marsilia Groza, Ciprian Dobre, Florin Pop, "An integrated architecture for future studies in data processing for smart cities", Microprocessors and Microsystems, Elsevier, vol. 52, p. 335-342, 2017

4) ***Journal 1.1 IF***: **Cristian Chilipirea**, Andreea-Cristian Petre, Ciprian Dobre, Florin Pop, George Suciu, "A simulator for opportunistic networks", Concurrency and Computation: Practice and Experience, Wiley Online Library, vol. 29, nr. 2, p. e3814, 2017

5) Andreea-Cristina Petre, **Cristian Chilipirea**, Mitra Baratchi, Ciprian Dobre, Maarten van Steen, "WiFi tracking of pedestrian behavior", Smart Sensors Networks, Elsevier, p. 309-337, 2017, **(Chapter 2 is based on this work)**

6) Vlad Ioan Nistorica, **Cristian Chilipirea**, Ciprian Dobre, "How many people are needed for a crowdsensing campaign?", 12th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP), p. 353-358, 2016

7) ***More than 10 citations***: **Cristian Chilipirea**, Andrei Ursache, Dan Octavian Popa, Florin Pop, "Energy efficiency and robustness for IoT: building a smart home security system", 12th IEEE international conference on intelligent computer communication and processing (ICCP), p. 43-48, 2016

8) ***Workshop A* conference***: **Cristian Chilipirea**, Alexandru Constantin, Dan Popa, Octavian Crintea, Ciprian Dobre, "Cloud Elasticity: going beyond demand as user load", Proceedings of the Third ACM International Workshop on Adaptive Resource Management and Scheduling for Cloud Computing (ARMSCC) of ACM Symposium on Principles of Distributed Computing (PODC), p. 46-51, 2016

9) ***More than 10 citations***: **Cristian Chilipirea**, Andreea-Cristina Petre, Ciprian Dobre, Maarten van Steen, "Presumably simple: monitoring crowds using WiFi", 17th IEEE International Conference on Mobile Data Management (MDM), vol. 1, p. 220-225, 2016, **(Chapter 3 is based on this work)**

10) **Cristian Chilipirea**, Ghita Laurentiu, Mirona Poescu, Sorin Radoveanu, Vladimir Cernov, Ciprian Dobre, "A comparison of private cloud systems", 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA), p. 139-143, 2016

11) **Cristian Chilipirea**, Andreea-Cristina Petre, Ciprian Dobre, "Big Data Uses in Crowd Based Systems", Resource Management for Big Data Platforms, Springer, p.441-459, 2016

12) **Cristian Chilipirea**, Andreea-Cristina Petre, Ciprian Dobre, Maarten van Steen, "Proximity graphs for crowd movement sensors", 10th IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), p. 310-314, 2015

13) **Cristian Chilipirea**, Andreea-Cristina Petre, Ciprian Dobre, Maarten van Steen, "Filters for wi-fi generated crowd movement data", 10th IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), p. 285-290, 2015, **(Chapters 2 and 3 are based on this work)**

14) Adriana Draghici, Taygun Agiali, **Cristian Chilipirea**, "Visualization system for human mobility analysis", 14th IEEE RoEduNet International Conference-Networking in Education and Research (RoEduNet NER), p. 152-157, 2015

15) **Cristian Chilipirea**, Andreea-Cristina Petre, Ciprian Dobre, Florin Pop, George Suciu, "A simulator for analysis of opportunistic routing algorithms", 14th IEEE International Symposium on Parallel and Distributed Computing, p. 27-36, 2015

16) ***Journal 4.3 IF - Q1***: **Cristian Chilipirea**, Andreea-Cristina Petre, Ciprian Dobre, Florin Pop, "Enabling mobile cloud wide spread through an evolutionary market-based approach", IEEE Systems Journal, vol. 10, nr. 2, p. 839-846, 2015

17) **Cristian Chilipirea**, Andreea Petre, Ciprian Dobre, Florin Pop, Fatos Xhafa, "Enabling vehicular data with distributed machine learning", Transactions on Computational Collective Intelligence XIX, Springer, p. 89-102, 2015

18) Andreea-Cristina Petre, **Cristian Chilipirea**, Ciprian Dobre, "Delay tolerant networks for disaster scenarios", Resource Management in Mobile Computing Environments, Springer, p. 3-24, 2014

19) **Cristian Chilipirea**, Andreea-Cristina Petre, Ciprian Dobre, "Social-based routing algorithm for energy preservation in mobile opportunistic networks", International Journal of Embedded Systems, Inderscience Publishers Ltd, vol. 6, nr. 1, p. 14-27, 2014

20) **Cristian Chilipirea**, Andreea-Cristina Petre, Ciprian Dobre, "Predicting encounters in opportunistic networks using gaussian process", 19th IEEE International Conference on Control Systems and Computer Science, p. 99-105, 2013

21) ***More than 10 citations***: **Cristian Chilipirea**, Andreea-Cristina Petre, Ciprian Dobre, "Energy-aware social-based routing in opportunistic networks", 27th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA), p. 791-796, 2013

22) ***More than 10 citations***: Mihaela-Catalina Nita, **Cristian Chilipirea**, Ciprian Dobre, Florin Pop, "A SLA-based method for big-data transfers with multi-criteria optimization constraints for IaaS", 11th IEEE RoEduNet International Conference, p. 1-6, 2013