UBIQUITOUS PRECISE TRACKING: FROM ACTIVITY DETECTION OVER INDOOR TRACKING, TO OUTDOOR VEHICLE POSITIONING

by

MOHAMED IBRAHIM AHMED

A dissertation submitted to the

Graduate School–New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Marco Gruteser

and approved by

New Brunswick, New Jersey

January, 2021

ABSTRACT OF THE DISSERTATION

Ubiquitous Precise Tracking: From Activity Detection over Indoor Tracking, to Outdoor Vehicle Positioning

By Mohamed Ibrahim Ahmed

Dissertation Director:

Marco Gruteser

Context awareness and tracking have changed our daily activities and style of living over the last three decades. Many applications have fueled research and industry efforts to establish accurate, energy-efficient, yet scalable and easy to deploy tracking and sensing systems. One of these applications is indoor and outdoor energy saving. For example, precise room occupancy estimation and activity sensing enable better control of indoor amenities such as light, heating, and air conditioning. These indoor applications can be extended to outdoor use for smart cities, e.g., automatic decrease of lighting for empty streets. However, current tracking systems can still not meet the aforementioned stringent requirements, and as a result there are no real world deployments of such applications. For example, conventional wireless tracking relies on WiFi received signal strength, and more recently on channel state information (CSI) offering decimeter level tracking accuracy. However, these tracking systems require either extensive fingerprints collection (wardriving), the knowledge of anchor locations and/or require expensive hardware that prevents wide deployment of such systems. Therefore, these applications with their sensing requirements still demand more accurate and scalable solutions.

This thesis focuses on developing wireless tracking solutions targeting submeter accuracy indoors and meter-level accuracy outdoors by leveraging unconventional wireless signals including visible light and WiFi Fine Time Measurements (FTM). These tracking algorithms can adaptively learn and simultaneously map the environment/anchors while tracking users.

The goal of this research is to propose tracking and context aware sensing systems that can tweak their parameters and map the environment through crowd-sourcing without the need of offline training. In particular, the proposed solutions include: (i) EyeLight, a device-free sensing system based on visible light to enable accurate tracking indoors and provide occupancy estimation, room activity recognition services; This system integrates photo-sensors with light bulbs easing its deployment compared to existing systems requiring the deployment of photosensors on the floor, (ii) an open platform for experimenting with WiFi fine time measurements and a general, repeatable, and accurate measurement framework for evaluating time-based ranging systems, (iii) Wi-Go, a system that simultaneously tracks vehicles and maps WiFi access point positions by fusing WiFi FTMs, GPS, and odometry information. We believe these three systems enable energy-efficient, continuous, precise and easy to deploy indoor and outdoor tracking and context awareness sensing solutions.

Acknowledgements

This dissertation would not have been possible without the help and support of a large group of wonderful people to whom I owe a lot of gratitude, of course after God.

First, I would like to express my deepest gratitude to my advisor and my mentor, Professor Marco Gruteser. During six year, he was like a big brother to me, and always was been there to help me. Because of him, my PhD journey was very smooth, despite facing many research and life difficulties. He gave me the freedom to work on a wide range of research project. I deeply feel fortunate for having you as my advisor, and I could not wish for a better advisor.

Second, I am really grateful for Professor Richard Howard who was like a second advisor to me. He has supported me and guided me through a lot problems. I truly admire and respect him.

Third, I would also like to thank Prof. Dimitri Metaxas, Prof. Badri Nath, Prof. Richard Martin, and Prof. Romit Roy Choudhury, my qualifier and dissertation committee members, for accepting my invitation, the insightful discussions and their constructive comments.

Fourth but foremost, I dedicate this thesis to my deceased parents, Ibrahim, and Khadiga. I was really hoping that they can live till that moment. Without their continuous support and love, I would not even have left Egypt and started my PhD journey here at Rutgers. Also, I would like to thank my wife, Rowaida, who left everything in Egypt to join me in my challenging journey here. Pursing a PhD while raising my children was challenging for me and my wife, but at the same time, it was a lot of fun. I would like also to thank my sister Marwa, my brother Ahmed, my father-in-law Mohamed, my mother-in-law Manal, my cousin Ayman and my aunt Nona. My family was always feeding me the strength and support to carry on and accomplish my career goals.

Finally, I am thankful for being part of WINLAB and for having all my dear friends and collaborators there including Viet, Siddharth, Minitha, Hansi, Ali, Murtada, Hongyu, Mohsen, Luyang, Gorkem, Bin, Shubham, Cagdas, Tam, Ashwin, Michael, Ivan, Noreen and Elaine. I would like to thank my collaborators at General Motors Research, Bo Yu and Fan Bai.

Dedication

To my deceased parents, Ibrahim & Khadiga, always in my heart

my wife, Rowaida

and my children, Mariam, Sarah & Ali

Table of Contents

Abstract ii

Acknowledgements iv

Dedication vi

List of Tables ix

List of Figures x

1. Introduction 1

1.1. Motivation 1

1.2. Existing literature on Active and Passive Tracking 2

1.3. Research Constraints 5

1.4. Design Guidelines 5

1.5. Contributions 6

2. Light-and-shadow-based Occupancy Estimation and Room Activity Recognition 8

2.1. Introduction 8

2.2. Background and Related Work 11

2.3. EyeLight design 13

2.4. Tracking Algorithms 17

2.5. Room Activity and Occupancy Recognition 21

2.6. EyeLight prototype and testbed 23

2.7. EyeLight evaluation 24

2.7.1. Light barrier crossing detection accuracy 24

2.7.2. Localization error 25

2.7.3. Room Activity Recognition and Occupancy Estimation 27
2.7.4. Microbenchmark experiments 28
2.8. Discussion and Conclusion 30

3. Verification: Accuracy Evaluation of WiFi Fine Time Measurements on an Open Platform 32
3.1. Introduction 32
3.2. Background 34
3.2.1. Multipath Challenge 34
3.2.2. Evaluation Challenges 35
3.2.3. Fine Time Measurement 36
3.3. Open WiFi Time-of-Flight Platform and Basic Ranging Calibration 37
3.3.1. Open WiFi FTM Tool 37
3.3.2. Experimental Setup 39
3.3.3. Basic Ranging Accuracy Calibration 40
3.4. Experimental Framework 43
3.4.1. Sampling Effect 44
3.4.2. Multipath Effect 45
3.5. Correcting Ranging Errors 50
3.5.1. Temporal Filtering 51
3.5.2. Spatial Filtering 53
3.5.3. Evaluation 53
3.6. Related Work 56
3.7. Conclusion 57

4. Wi-GO: Accurate and Scalable Vehicle Positioning using WiFi Fine Timing Measurement 58
4.1. Introduction 58
4.2. Background 61
4.3. Can WiFi Augment GPS? 63

4.4. The Wi-Go System . . . 65
4.4.1. System Overview . . . 65
4.4.2. Mobile Multilateration . . . 66
4.4.3. Uncertainty Weighting . . . 67
4.4.4. Congestion-Aware Adaptation of FTM Request Transmission . . . 68
4.5. The FTMSLAM Algorithm . . . 71
4.5.1. Location Modeling and Initialization . . . 71
4.5.2. Opportunistic Bearing Estimation . . . 72
4.5.3. FTMSLAM Tracking Framework . . . 73
4.6. Parked Vehicles as Pseudo APs . . . 75
4.7. Evaluation . . . 77
4.7.1. Experimental Setup . . . 77
4.7.2. Urban Canyon Evaluation . . . 77
4.7.3. Residential Apartments Evaluation . . . 82
4.7.4. Micro Benchmarks . . . 82
4.7.5. Evaluating Adaptive FTM Rate . . . 84
4.7.6. Discussion . . . 85
4.8. Related Work . . . 86
4.9. Conclusion . . . 87
5. Conclusion . . . 88
5.1. Lessons Learned . . . 89
5.2. Future Directions . . . 89
References . . . 91

List of Tables

2.1. Room activity and occupancy categories 23

3.1. Comparing the average ranging accuracy for using different WiFi chipsets as the AP, while fixing the STA using WiFi card A, under different bandwidth and band configurations. The average ranging accuracy is reported in meters for 1 meter actual distance. 38

4.1. Median tracking error of related technologies in Manhattan, NY. 64

4.2. AP localization mean error in Manhattan experiments. 78

4.3. Wi-Go Latency. 79

4.4. Simulation configuration. 80

4.5. AP localization error in the residential apartment complex experiment. . . . 82

List of Figures

2.1. Overview diagram of components in EyeLight. 14

2.2. Receiver. 16

2.3. Raw readings from one receiver. 18

2.4. Virtual light barrier crossing detection. 20

2.5. EyeLight testbed. There are 6 light nodes with distance between adjacent pair is 2.5m. The room has a central table, a number of chairs, and a projector screen. 24

2.6. The distribution of different activities and occupancy categories in the dataset. 25

2.7. True positive rate and false positive rate of delta detection algorithm for all different sensors. 26

2.8. CDF of localization error for three cases: 1) using only spikes detection, 2) using only delta detection, and 3) combined detection. 26

2.9. Confusion matrix for activity identification in a conference room. Each value in the matrix is rounded to two decimal places. Total size of the dataset is 102889 feature vectors, corresponding to 28.5 hours. 27

2.10. Confusion matrix for occupancy estimation in a conference room. Each value in the matrix is rounded to two decimal places. Total size of the dataset is 1710 feature vectors, corresponding to 28.5 hours. 27

2.11. (a) Delta values for different distance between two nodes. (b) Location median error for different number of nodes. (c) Delta values for different ambient light settings. (d) Delta values for different types of floor carpets.(e) Types of carpet in (d). (f) Effect of lamp shade. 29

3.1. Multipath problem. 35

3.2. FTM Protocol Overview. 37

3.3. Outdoor open space. This setup uses the WiFi card A for both STA and AP sides. 40
3.4. Maximum range in outdoor open space (1.4 m height, offset corrected. This setup uses the WiFi card A for both STA and AP sides. 41
3.5. Outdoor open space grass ground (0.7 m height). Comparing ranging results for using WiFi card A as AP compared to AP B while using WiFi card A as STA in both scenarios. 41
3.6. Effect of varying number of samples per burst. 44
3.7. Blocking LoS reception (1.4 m height, offset corrected). 45
3.8. Outdoor parallel to 7-floor building (1.4 m height, offset corrected). 46
3.9. Mobile environment effect for static sender and receiver. 46
3.10. Indoor scenario: room setting (0.7 m height). This setup uses WiFi card A for both STA and AP sides. 48
3.11. Indoor scenario: corridor (1.4 m height). This setup uses WiFi card A for both STA and AP sides. 49
3.12. (a) Indoor scenario (room (0.7 m height)) for comparing ranging results for using WiFi card A as AP compared to AP B while using WiFi card A as STA in both scenarios. (b) 1 cm step in indoor room (1.4 m height, offset corrected). Zero distance represents no signal. 50
3.13. Indoor AP outdoor STA. Both AP and STA use WiFi card A. Zero distance represents no signal 51
3.14. Indoor AP (AP B), outdoor STA (WiFi card A). Zero distance represents no signal. 51
3.15. Displacement limit illustration. 52
3.16. Effect of displacement limit on ranging error for indoor AP and outdoor STA 54
3.17. Illustration for the displacement limit and its effect of on the ranging and localization error along with a map for the indoor testbed. Red dots represents STA locations. 55

4.1. (a) WiFi edge over GPS. (b) Effect of linear movement of a vehicle on AP localization error. 62
4.2. WiFi FTM ranging accuracy. 64
4.3. Wi-Go System Architecture Overview. 65
4.4. Illustration of our Mobile Multilateration. A vehicle moves from point a, to b, then from b to c. 66
4.5. FTM Weighting. (a) Triangular inequality. (b) Assigning measurement confidence based on a time window. 68
4.6. Angle estimation illustration. 72
4.7. Vehicle tracking error of Wi-Go Vs current localization systems in Manhattan experiments. 78
4.8. (a) $25^{\text{th}}\%$, median, and $75^{\text{th}}\%$ latency for Wi-Go (adaptive spb) compared to baselines using ns-3 simulations, and (b) mean and standard deviation of the latency versus number of completed AP rangings in one AP list scan for 75% penetration ratio. 81
4.9. Evaluating vehicle tracking accuracy in the residential apartment complex experiment. 81
4.10. APs density effect on vehicle's localization error in Manhattan. 83
4.11. FTMSLAM algorithms effect on AP localization error. 83
4.12. Ranging correction effect on AP localization error. 83
4.13. Evaluating our adaptive FTM message rate approach leveraging FTM ranges only. 84

Chapter 1

Introduction

1.1 Motivation

Fine-grained localization is a crucial step toward a plethora of context awareness applications including activity detection, occupancy estimation, and autonomous driving. These applications can turn the dreams of smart homes and cities into reality. For example, submeter accuracy tracking indoors can enable a smart home to detect empty rooms and hence, adapt amenities such as light, heating, and air conditioning. A better control of indoor amenities can save energy without the need of user intervention. Moreover, precise tracking can save lives by helping emergency responders to locate persons stuck behind walls. Precise tracking can also lead to automatic detection of fainted persons, and quickly reviving them (e.g. a heart attack, seizure, or low blood sugar). These indoor applications can be extended to outdoor use for smart cities, e.g., automatic decrease of lighting for empty streets. A continuous uninterrupted tracking can enable contact tracing and social distancing applications and hence help limit the spread of contagious diseases such as COVID-19.

Over the last three decades, researchers have shown that precise tracking is achievable, however, none of this work has been deployed widely into production, specifically indoors. With the stubborn challenge of identifying the direct path in multipath propagation, researchers proposed various solutions that either count on laborious fingerprinting, specialized/expensive prototypes and/or assuming the knowledge of anchor locations. While the GPS is the widely used localization system, it still suffers from performance degradation indoors and in urban canyons.

The research presented in this thesis addresses the above challenges in the context of both algorithms and systems in order to answer the following questions:

Can we achieve submeter tracking accuracy indoors and meter-level accuracy outdoors using visible light and WiFi FTM by adaptively learning the environment and mapping existing anchors while tracking the users?

This thesis answers the above questions by designing tracking algorithms, that adaptively learn and map the environment without the need of offline training (e.g. wardriving) or preprocessing (e.g. assuming the knowledge of anchor locations). Specifically, our tracking algorithms boot with inaccurate user location estimate, and leverage that to map the environment and consequently improve the user localization estimate. This approach is inspired by the robotics Simultaneous Localization and Mapping (SLAM) framework. Our localization algorithms can better learn the environment and improve the tracking accuracy by adaptively tweaking their parameters for every user visit. In other words, instead of performing localization in two separate phases: a) estimate anchors location, b) estimate users location, we estimate both anchors and users location simultaneously. In this case, the algorithms can adapt to changes in the environment or anchor locations, and still can benefit from the users traces to better estimate anchor locations.

In particular, this thesis provides an indoor tracking system, an open ranging tool, and outdoor tracking system. *First*, in indoor environments, we propose a system using ceiling lights to track human presence and infer room activities without the need to carry a device or deploy photosensors on the floor. *Second*, an open ranging tool to experiment with WiFi FTM and a repeatable measurement framework for time-of-flight ranging systems. *Third*, a vehicle and WiFi access points (APs) localization system that fuses WiFi FTM, GPS, and odometry information. We believe these systems facilitate meter-level tracking while counting on either off-the-shelve WiFi APs or low cost, plug and play light bulb sensing prototype.

1.2 Existing literature on Active and Passive Tracking

Current activity sensing and tracking techniques are either based on active or passive sensing. In active sensing, the system tracks the user through a device which she carries. For example, a user carries a phone which receives and transmits wireless signals. On the other hand, passive systems sense the presence and effect of a user on the signal being transmitted and received by existing devices in the surrounding environment. For example, light curtains can detect user presence when a user crosses and blocks the infrared signal transmitted between a transmitter and a receiver.

Over the last three decades, the research in tracking has mainly focused on improving the tracking accuracy going from tens of meters to meter-level accuracy outdoor, and from meters to decimeter-level accuracy indoor. However, usually this huge improvement trades accuracy with scalability and complexity. Therefore, these efforts still face major challenges to find their ways to production.

In this section, we review these two major categories of tracking research work, *active tracking* and *passive tracking*, identify missing requirements that this dissertation aims to address.

Active Tracking. Active tracking counts on tracking a device the user is carrying through wireless sensing. Starting from Active Badge [1], that transmits a unique code on IR signals to sensors placed at known locations, to Active Bat [2] that transmits ultrasonic signals to grid of receivers. These receivers estimate the distance to the transmitter using the time-of-flight of these ultrasonic signals. Similarly, the Cricket system [3] leverages preinstalled nodes that transmits ultrasound and RF signals, while the user carry mobile devices that hear and estimate distance to these fixed nodes. These tracking systems leverage multilateration to estimate the location of the device carried by the user. In multilateration, a localization system estimates the user location ideally by intersecting spheres around the anchor nodes. Therefore, such systems assume the knowledge of anchors location and they only need to estimate the distances between a device and each anchor node. Numerous research work [4–7] focused on improving the tracking accuracy reaching decimeter-level localization using time-of-flight or angle-of-arrival measurements over MIMO WiFi access points (with at least 3 antennas and require access to CSI information) or specialized hardware such as WARP. Moreover, this research work requires the knowledge of WiFi APs location. On the other hand, RADAR [8] introduced the RF fingerprinting technique in which an offline phase is required to collect received signal signature labeled with its location and save this information to a database (radio map). In the online phase, RADAR uses a nearest neighbor technique to search in the database and pick the location that best matches the observed signal strength vector. A line of research work has extended the notion of RF fingerprinting by proposing different machine learning techniques to improve localization accuracy and complexity. Starting from proposing probabilistic techniques such as [9, 10] to recently the usage of deep learning for indoor localization [11]. These techniques have been extended to outdoor localization counting on cellular and WiFi signals [12, 13].

What are currently missing in device-based localization are techniques that simultaneously track users and off-the-shelf anchor nodes without the need of an offline phase for radio map building or anchors location estimation. Resolving these requirements can push the localization research into production through scalable and easy-to-deploy localization system. We propose Wi-Go that satisfy these requirements: a vehicle localization system that simultaneously tracks vehicles and WiFi off-the-shelf access points without assuming the knowledge of these APs or requiring an offline data collection phase. Angle-of-arrival techniques can complement our approach depending on the availability of at least 3 antennas.

Device-free Passive Tracking. Another category of tracking is the device-free passive localization [14], in which the tracked object do not carry a device or participate in the tracking process. For example, object tracking through computer vision [15–17], which requires the object to be in line-of-sight to preinstalled camera. In wireless sensing, systems were inspired by the radar community work, to detect and track humans behind walls using specialized hardware on Ultra-wideband and WiFi signals [18–21]. More recently, Wifi-based activity sensing (e.g., [22]), has been proposed, which generally achieves large coverage at lower accuracy and faces more challenges to scale to buildings with many occupants. Through visible light sensing, StarLight [23] can reconstruct fine-grained user skeleton postures in an office room equipped with LED lights on the ceiling and photosensors on the floor.

Although, existing device-free tracking achieves decimeter-level accuracy, a system, that is privacy-preserving, easy-to-deploy, and low cost and yet accurate, is still missing. We present a smart indoor environment based on existing indoor light infrastructure. We envision that a smart light bulb with visible light sensing capabilities can be plugged-and-played to existing light infrastructure.

1.3 Research Constraints

To realize our research goal of scalable and accurate tracking, we faced the following challenges:

Meter and submeter tracking. These tracking systems should be able to achieve submeter-level localization accuracy indoor and meter-level accuracy outdoor in order to enable applications such as indoor room level activity detection, outdoor autonomous driving.

Scalability and deployability. Achieving even millimeter-level tracking accuracy is possible, however, such existing accurate systems either require offline calibration/training process, the knowledge of anchor nodes or require specialized/high cost infrastructure resulting in not scalable nor easy-to-deploy solutions. Therefore, our tracking system should not require offline training/preprocessing process, and has to count on low-cost infrastructure.

Low-resolution sensing. High-resolution sensing such as indoor tracking through cameras raises privacy concerns, particularly in residential areas. Indoors, we leverage visible light low-resolution sensing particularly, few photosensors per light bulb compared to several mega sensors per camera. Accurate tracking and activity detection are challenging through such low-resolution data.

1.4 Design Guidelines

In order to satisfy the above constraints, we follow the following design guidelines:

Range-based localization. In order to achieve, scalable and accurate tracking, we choose to follow multilateration-based localization over fingerprinting-based localization. This design choice requires that an existing infrastructure can either estimate distance to a user's device or detect presence of a user in its vicinity. We avoid localization solutions that require laborious offline phase of wireless signature collection and ground truth labeling.

Crowd-sourcing SLAM. It may not be practical to assume that we know the locations of anchor nodes such as WiFi access. It is also not scalable nor robust to assume that we can estimate APs location through offline war-driving such as camera/LiDAR assisted robot scanning [24]. For example, a user can move AP to different location after such offline mapping procedure. Therefore, we design a localization system that incrementally improves tracking of users and WiFi APs through crowd-sourced simultaneous localization and mapping. On the other hand, ceiling light infrastructure has specific set of locations limited to ceiling power outlets. Therefore, even if the maps of these ceiling power outlets are not available, a LiDAR assisted robot or a user (e.g. recent phones such as iPhone 12 has LiDAR) can map the ceiling light infrastructure.

Unconventional sensing modalities. We leverage existing, yet unconventional wireless signals like visible light and WiFi Fine Time Measurements counting on already deployed and highly dense light and WiFi infrastructure. Specifically, we use over-the-shelf WiFi APs supporting WiFi Fine time measurements and design low-cost play-and-play smart light bulbs that can sense users in its vicinity.

1.5 Contributions

In this thesis, we focus on designing an accurate and scalable indoor and outdoor localization systems. In particular, this thesis hypothesizes that existing wireless signals such visible light and WiFi Fine Time Measurements can achieve decimeter-level indoor tracking and meter-level outdoor tracking through adaptive SLAM algorithms. These algorithms can map the environment/anchors while simultaneously track users. In this section, we shed the light on three pieces of work in this thesis that enables our vision of indoor and outdoor tracking.

Light-and-shadow-based Occupancy Estimation and Room Activity Recognition [25]. This work proposes EyeLight, an device-free indoor tracking system embedded in indoor lighting environment to sense the human occupancy and room activities. Instead of deploying light sensors on the floor, Eyelight integrates photosensors with light bulbs and exploits the light reflected off the floor to achieve an entirely device-free and light source based system. Sensor readings are fed into indoor tracking algorithms that adaptively learn the light levels in the room and tweak their parameters accordingly. Moreover, we feed these sensor readings to occupancy and activity recognition classifiers. We evaluate the performance of our system in terms of localization, occupancy estimation and activity classification, and find a 0.89m median localization error as well as 93.7% and 93.78% occupancy and activity classification accuracy, respectively. EyeLight will be presented in Chapter 2.

An Open Experimental Platform of WiFi Fine Time Measurements [26]. This work introduces an open platform for experimenting with WiFi fine time measurements and a general, repeatable, and accurate measurement framework for evaluating time-based ranging systems. We analyze the key factors and parameters that affect the ranging performance and revisit standard error correction techniques for WiFi time-based ranging system. The results confirm that meter-level ranging accuracy is possible as promised, but the measurements also show that this can only be consistently achieved in low-multipath environments such as open outdoor spaces. This work is detailed in Chapter 3

Accurate and Scalable Vehicle Positioning [27]. Driver assistance and vehicular automation would greatly benefit from uninterrupted lane-level vehicle positioning, especially in challenging environments like metropolitan cities. In this work, we show that WiFi Fine Time Measurements can complement current GPS and odometry systems to achieve lane-level positioning in urban canyons. We propose Wi-Go, a system that simultaneously tracks vehicles and maps WiFi access point positions by coherently fusing WiFi FTMs, GPS, and vehicle odometry information together. To prevent WiFi congestion, Wi-Go adaptively controls the FTM messaging rate from clients while maximizing the tracking accuracy. Wi-Go achieves lane-level vehicle positioning (1.3 m median and 2.9 m 90-percentile error), an order of magnitude improvement over vehicle built-in GPS, through vehicle experiments in the urban canyons of Manhattan, New York City, as well as in suburban areas (0.8 m median and 3.2 m 90-percentile error). Wi-Go is presented in Chapter 4.

Chapter 2

Light-and-shadow-based Occupancy Estimation and Room Activity Recognition

2.1 Introduction

Building-wide occupancy detection and activity sensing promises to enable a new class of applications across smart homes, elderly care, and retail marketing. In smart homes, for example, it could enhance control of lighting, heating, ventilation, and air conditioning based on sensed and predicted activities across rooms. Useful information ranges from basic occupancy and movement tracking to activity inference (e.g., sleeping, cooking, eating, watching TV or media). In elderly care, activity sensing allows quick detection of emergencies or changes in routine. In stores and showrooms, foot traffic statistics for individual aisles or product display areas are invaluable for ad placement and arranging products.

Existing occupancy sensing technologies. These activities are currently detected by a number of dedicated sensing systems, with Infrared (IR) motion sensing being especially prevalent. Passive or Pyroelectric Infrared (PIR) sensors detect the radiated IR energy from humans and animals [28]. However, PIR sensors require line-of-sight coverage, which increases the number of required sensors to cover a certain area. For example, previous work [29] required one sensor per 4 meter square area. A combination of environmental sensors such as CO2 , temperature, humidity, and pressure is used for occupancy estimation [30, 31]. PIR and these environmental sensors are also sensitive to other heat sources (e.g., hot appliances, sunlight and open window), and they are designed to detect movements, not presence, which limits its tracking of stationary users. For more fine-grained detection in a small area, light barriers detect motion when transmission between an IR transmitter and receiver is obstructed. Other device-free solutions have relied on cameras [32]. Although they are effective and ubiquitous in public places, cameras raise privacy issues, especially in residential areas. More recently, Wifi-based activity sensing (e.g., [22]), has been proposed, which generally achieves large coverage at lower accuracy and faces more challenges to scale to buildings with many occupants. Besides such device-free sensing, other approaches leverage user devices like smart watches and smart phones (e.g., [33]). The disadvantage of these approaches is that users need to continuously carry, wear, and usually charge them.

More recently, fine-grained localization and activity sensing using visible light has been investigated. Current VLS work mainly uses active techniques (users are required to carry sensors or devices) and focuses on line-of-sight communication between transmitter and receiver [34, 35]. Among passive (device-free) techniques, LiSense [36] demonstrates fine-grained gesture and human skeleton reconstruction using visible light sensing but requires deploying photodiodes on the floor to obtain line-of-sight links with the transmitters. CeilingSee [37] converts ceiling mounted LED luminaries to act as photosensors, to infer indoor occupancy, but requires dense deployment (1.25m between nearby pair) of LED luminaries because of reduced sensitivity of LEDs acting like photosensors compared to dedicated photosensors. None of these technologies can therefore provide device-free occupancy sensing beyond line of sight, which would enable building scale fine-grained activity sensing with lower deployment overhead (i.e. using fewer sensors).

EyeLight Approach. We introduce EyeLight, a device-free occupancy detection and activity sensing system exploiting opportunistic, indirect light sensing so that it can be integrated in a set of networked LED light bulbs. EyeLight forms a mesh of virtual light barriers among nearby light bulbs to sense human presence as they move across the room. Exploiting light provides attractive properties. Due to its nanometer wavelength it is highly sensitive to small motion and objects when compared to RF waves. Also, unlike most RF techniques, light does not suffer from RF interference and cannot penetrate through walls, which preserves privacy and makes it easier to determine in which room an activity occurred.

Contrary to conventional light barriers, however, no direct line-of-sight is needed—the system exploits opportunistic reflections in the environment (e.g., shadows and reflections off the floor). Indirect tracking of users based on their shadows, enlarges the system's operation range, compared to line-of-sight based solutions like PIRs. This allows covering a space with fewer sensors and provides more freedom in deployment locations, making it easier to reuse infrastructure that already exists (for example, recessed can lighting where power is available but, due to the recessed location, line-of-sight may not exist to the entire space). Such reuse allows for building-scale motion tracking and activity sensing with little installation overhead (no additional building wiring is needed).

The prototype design makes use of the trend of LED light bulbs increasingly containing electronics and having access to plentiful power. Light bulbs are integrated with photosensors and networked to coordinate signaling and to upload sensor data for processing. We design barrier crossing detection as well as occupancy and activity classification algorithms based on sensed changes in the reflected light levels, for example, due to a shadow. This work significantly extends prior work [38] by 1) using dual purpose signaling light (illumination without causing flicker to the eyes while sending the signature of the node), 2) a room-scale prototype with localization and activity recognition, as well as 3) enhancing sensitivity to operate on different reflective surfaces and longer sensing distances (up to 3 meters).

In summary, the major contributions of this work are as follows:

- exploring the feasibility of creating opportunistic meshes of virtual light barriers between modified light bulbs by exploiting reflections off room surfaces.
- proposing a sensitive photoreceiver design for lamp-based light barriers that can detect light reflected from different room materials, including dark floor carpet.
- designing light-based occupancy tracking and room activity recognition algorithms and exploring their potential when deployed across a room's ceiling lighting system.
- designing and implementing a room-scale prototype system and evaluating EyeLight in terms of localization accuracy, estimating occupancy, and recognizing different room activities based on 28.5 hours of recorded data.

2.2 Background and Related Work

Visible light sensing can be implemented directly in illumination systems. Adoption of LED lighting is growing rapidly [39] due to their 75% lower energy consumption and 25 times longer lifetime than incandescent lighting. LEDs can also be switched faster than incandescent and fluorescent light sources, which allows rapid signaling with light sources and enables novel applications [40]. Given the presence of solid state devices and power converting circuits (AC to DC) in LED light bulbs, it has also become easier to integrate additional electronics in such devices, particularly since power is plentiful. To be acceptable, signaling between lights usually has to be imperceptible for human observers.

Human light perception. Imperceptible signaling is possible because human eyes respond slower than photodiodes to light changes. The *critical flicker frequency (CFF)* [41], typically 100Hz, defines the frequency beyond which our eyes cannot perceive time-variant light fluctuation and see only its average luminance. This effect is similar to a low pass filter with the CFF as cut-off frequency. While the exact frequency depends on other factors (such as light intensity, color contrasts, etc.,) sufficiently fast signaling can surpass the flicker perception of human eyes, yet still remain detectable by photosensor front-ends.

Our eyes also perceive light intensity logarithmically, instead of relatively linearly like photosensors. Therefore, a small change of light intensity that is perceivable in a dark room can be invisible in a brighter room. A photosensor calibrated for this range of light levels can easily detect such differences, however.

Existing passive sensing techniques. A major approach to occupancy sensing is using RF signal measurements, based on RSSI ([8, 42]) or time-of-flight [43]. Cameras are also used for monitoring people indoors, but they raise privacy concerns [44]. Other approaches, including Capacitance [45] and Pressure [46] require sensors on the floor, which is not practical for installation in several cases.

Light, both visible and infrared, has long been used for motion detection. Light barriers or curtains [47, 48], for example, detect when a light beam between a source and a photosensor is blocked by a moving object. Since light beams can be easily focused through lenses, they allow more precise movement detection than radio frequency sensing. To ease deployment, retro-reflective sensors package the light source and sensor into a single device but this usually requires a retroreflector that is carefully aligned to reflect the light back to the sensor.

Visible Light ([34, 49]), an emerging short range communication technology, has been recently explored for indoor localization applications, thanks to the growing use of LED bulbs. More recent works [36, 37] explored the use of ceiling lights in the visible light spectrum to track people indoor. However, either the photosensors are deployed on the floor to achieve line-of-sight to the ceiling lights, which significantly complicates the deployment, or the LEDs are forward biased to function as light sensors, which leads to lower sensitivity and small coverage in the line-of-sight area.

Challenges in reflective light sensing. Is it possible to achieve both large coverage and ease of deployment by forming a mesh of opportunistic reflective light barriers?. Allowing for indirect, reflective light sensing could extend the sensing range, since movement can be detected not only directly in line-of-sight of a sensor but also anywhere along the longer reflected path of a light signal. Eliminating the line-of-sight constraint also provides more freedom in placing the lights and sensor. In particular, this approach would allow integrating all necessary components into light bulbs, which would significantly simplify the deployment process: the system could be installed by simply changing light bulbs. Note also, that power requirements of the added electronics are met by the power source to the LED light and does not require any battery or additional wiring.

This approach introduces several challenges, however. *First*, the detector now has to recognize much weaker light levels due to two reasons: 1) received light power decreases proportional to square of the distance and reflected paths tend to be longer (for example, the distance with a floor reflection to an adjacent ceiling light is more than double compared to the distance with photosensors directly on the floor), and 2) most typical room surfaces absorb or diffuse a substantial part of the light (e.g. a dark carpet), thus the incident light power on the photodiode is reduced. *Second*, the reflected paths are less well defined. The exact path depends on the position and shape of objects in the space and it is possible that the light reaches the photosensor along multiple paths (akin to radio multipath effects). Motion tracking, occupancy estimation, and activity detection algorithms have to be robust to such effects. *Third*, the receiver should be able to distinguish light from different sources. In addition, any signaling technique used for this purpose should remain imperceptible and not detract from the illumination function of light bulbs.

A common method for detecting a weak signal is a correlation detector with a known pseudorandom number (PRN) sequence. This effectively spreads the signal bandwidth leading to a significantly enhanced signal to noise ratio. Applying this to EyeLight is challenging, however. First, achieving high processing gains requires long PRN sequences[1]. Given the limited modulation rate of high power lighting LEDs, these sequences would take seconds to minutes to transmit, which is longer than the duration of human movement events that we seek to detect. Second, transmitting continuous PRN sequences with on-off keying would halve the brightness of the ceiling lights, since one can expect equal number of on and off symbols. Third, as a result of the spectrum spreading property, PRN sequences introduce low frequency components, which increases the chance of flicker for human eyes.

2.3 EyeLight design

EyeLight realizes an opportunistic mesh of reflected light barriers through synchronized signaling from networked transmitters and a pulse-based power measurement technique based on sensitive receiver hardware. It relies on modified LED light bulbs to transmit modulated light and contains sensitive photodetectors to detect light signals. It coordinates signaling among light sources so that a virtual light barrier can be established between nearby pairs of lights without interference from other participating light sources. These light barriers are opportunistic since the light needs to reflect off surfaces in the environment to reach the photodetector on an adjacent light bulb. The key rationale for integrating both signaling and sensing components in light bulbs is that it reduces installation and maintenance costs, as power is already available at the lamps.

It addresses the challenge of invisible modulation of LED light bulbs together with self-interference free detection sensitive enough to measure weak reflections through a synchronous, pulse-based power measurement technique. Bulbs emit a periodic pulse, which is

[1]For example, GPS system uses 1023-bit PRN sequence which repeats itself every 1ms.

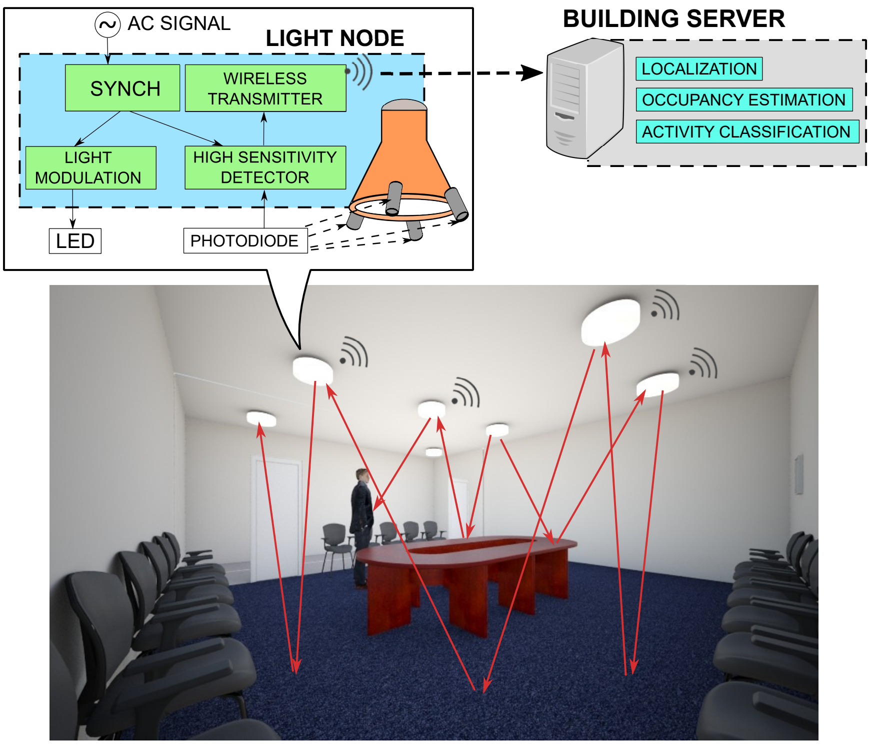


Figure 2.1: Overview diagram of components in EyeLight.

short enough to remain imperceptible, meaning it does not noticeably affect brightness of the light and does not cause flicker. Receivers measure the signal power of the pulse and compare it to the overall light level to track movement and changes in the room.

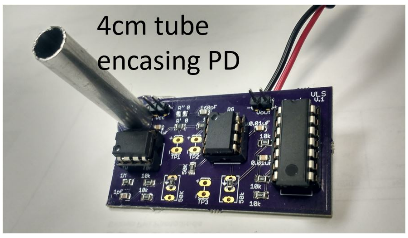
The light nodes have wireless connectivity to report their measurements to a server, where tracking and activity detection algorithms process the datastream to monitor the movements and activities of occupants. We assume that light bulbs can be mapped with their location in the room during installation. Self-localization algorithms may also be possible. Fig. 2.1 shows an overview of the components in EyeLight.

Transmitted Signal. The transmitted signal should allow the receiver to separate light emitted by one specific transmitter from other ambient light sources, while remaining imperceptible to the human eye. In theory, this can be achieved with straightforward ON-OFF signaling. Since flicker perception depends on frequency, this raises the question of whether the high power LEDs used in light bulbs can be switched fast enough to remain imperceptible. We measured the rise and fall time of an off-the-shelf LED bulb (Ecosmart 65W BR30) and observed that the lamp takes about 0.1ms to rise to 90% of its peak intensity and a shorter time to fall. This shows that the light bulbs are fast enough for ON-OFF signaling without introducing flicker to human eyes (previous research [41] showed that the critical flicker frequency of human eyes when perceiving a strong single light source is only about 100Hz).

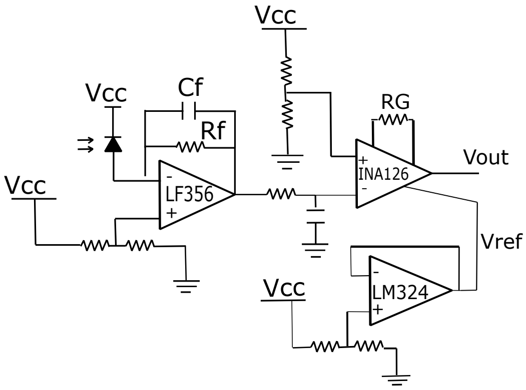
In addition to eliminating flicker, the signal also should not significantly affect the overall illumination level. We therefore use periodic signaling, which only occurs in a short slot out of a longer cycle. When ceiling lights are used to illuminate the space, the light would briefly switch off during its slot, while remaining on during the rest of a cycle. This design reduces the lamps' brightness by only a negligible amount. Conversely, when lights are off, the lamps could briefly switch on during their slot to signal. Our implementation focuses on the former. Supporting both modes would require additional calibration of receiver sensitivity.

Receiver. Sensing reflected light off the floor with photosensors deployed on the ceiling is a challenging task. The photsensor frontend needs a high sensitivity to receive weak light and fast response time to detect the modulated signal. These requirements are usually at odds with each other. We achieve these requirements by carefully designing a receiver circuit combining several components (Fig. 2.2(b)). Since we require a fast light sensor to detect short pulse (under 1ms) from the transmitter, we use a photodiode as our sensor. The weak current generated by the photodiode is amplified through a Transimpedance Amplifier. The amplifier acts as the current-to-voltage converter—it converts and amplifies the photocurrent generated by the photodiode to a voltage that can be read out. The amplifying gain of the TIA is set by the feedback resistor R_F following: $V_{out}/I_P = -R_F$.

Compared to a simple detector (a photodiode in series with a resistor R), the transimpedance amplifier has much faster response time than the time constant $R_F * C_d$ (with C_d is the internal capacitor of the photodiode). Therefore, we can use a larger value of R_F to increase the gain while maintaining fast response at the front end. However, the value of the feedback resistor cannot be arbitrarily large since it is limited by two factors: large Johnson thermal noise ($v_n = \sqrt{4k_B TR}(V)$) can reduce SNR of the frontend, and low input rolloff frequency ($f_{RCin} = \frac{1}{2R_F C_{in}}$) can limit our operating frequency. To further boost the gain, we use a second stage amplifier: an instrumentation amplifier (INA126). The output



(a) Fabricated Receiver.



(b) Receiver Circuit.

Figure 2.2: Receiver.

of the amplifier is given by,

$$V_o = G(V_+ - V_-) + V_{Ref}$$

where V_{Ref} is a reference voltage being fed to the instrumentation amplifier, and G is a controllable gain. One can consider the two inputs to the INA126 as output voltages from two arms of a Wheatstone bridge [38], whose difference we seek to amplify. The negative input V_- is fed with the output of the TIA, while the positive input V_+ is fed with a constant voltage from a voltage divider. Note that G and V_+ are two controllable factors that help the receiver adapt to different light levels.

Multiple Transmitters and Receivers. The previous two sections describe how a single pair of transmitter and receiver can communicate through reflected light on the floor. When multiple transmitters are in the room, each light node needs its own identification—when the sensing module detects a light level change because of a shadow, it needs to recognize which light source created that shadow. Therefore, each LED bulb needs a mechanism to send its own signature. This can be done in the frequency domain, as in [36], or time domain. We choose the time domain because of its simplicity when combined with synchronization from the common AC power signal, which our design assumes. As in other prior work [23], the main idea is that each light fixture chooses its own time slot, during which it signals.

For the time-slot based mechanism to work, the clocks of all light nodes need to be synchronized. We implement this by using the common 60Hz AC signal available from the mains power [50]. Recall that a key motivation for incorporating signaling and sensing into light bulbs was the easy availability of power. We therefore also assume a common AC signal for synchronization. Each zero-crossing event of the mains power signal marks the start of a *cycle* for EyeLight, making the cycle length half the period of the AC signal (about 8ms).

Given n light bulbs that can potentially observe signals from each other, the system requires n timeslots to uniquely assign a slot to each lamp, which lets the receiver identify the signaling lamp based on the current time. Note that, as in wireless systems, spatial reuse is possible and walls that block light make the reuse of slots across different lamps in a building even easier. This keeps the total number of required time slots relatively small. The maximum number of timeslots that can be supported is determined by the cycle length and the lower slot duration bound derived from the LED rise time.

Besides signaling, each node also looks for signal from other nodes through multiple receivers co-located with the LED lamp. The photosensors point to different directions to detecting signal from surrounding light nodes. For the sampling scheme, we employ a Round-Robin approach to maximize the number of samples per cycle: in each cycle, we let only one photosensor sample the light level in its view, then move to the next photosensors. This ensures each sensor has high enough sampling rate for detecting the fast signal from other nodes.

Fig. 2.3 shows an example of received light power at one receiver over consecutive cycles. This receiver is on node 2, so it observes a big dip in the second timeslot when node 2 signals. It also observes a smaller dip in the first timeslot, when the adjacent node 1 does signaling. This dip shows the effectiveness of our receiver design to sense weak reflected signals off the floor from an adjacent node.

2.4 Tracking Algorithms

The photodiode in each sensor converts the incident radiant energy P to the output photocurrent I_p, making our sensor a light power measurement device. In essence, our tracking

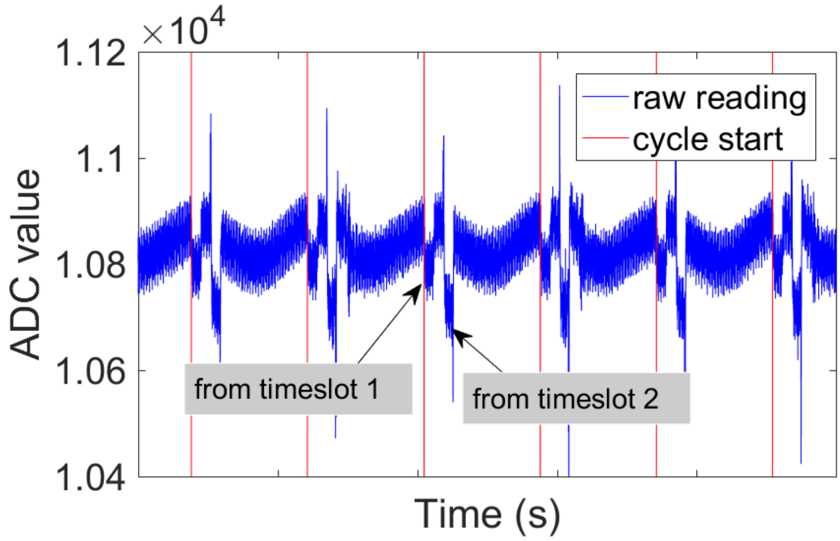


Figure 2.3: Raw readings from one receiver.

algorithms utilize signal power measurements over time, and compare them with the baseline light power level when the room is unoccupied. To improve the confidence of our localization, we introduce two methods, *Spike algorithm* for coarse-grained localization and *Delta algorithm* for fine-grained localization.

The first method measures if there is any change in received light power, which is caused by movement events surrounding the light node position. We detect this change by continuously taking average received power over an entire cycle for each sensor and using a threshold-based detection to detect when this average power deviates far away from base light level (when the room is empty). This approach, which we call *Spike algorithm*, only tracks movement at a coarse-level—it can only detect if there is a movement event in an area surrounding the spot on the floor a receiver is monitoring.

The second method aims at fine-grained level tracking—it determines whether a change occurred on a specific transmitter-receiver link. With multiple light nodes covering a room and each carrying several receivers, we can effectively create an opportunistic mesh of virtual light barriers to detect when a subject is passing by. Since each light source in the interference domain signals in a unique time slot, receivers can simply check for the presence of the ON-OFF signal in a particular time slot. If the signal can be detected the virtual light barrier is connected, otherwise it is interrupted. This technique is agnostic to most changes in ambient light level that can occur. Over time, the system can then monitor changes in the status of each link.

While the concept is intuitive, its implementation is challenging due to the complex light propagation environment. The system uses reflections off random surfaces rather than direct illumination or a special reflector as in a retro-reflective light barrier. This means that the light level change when the virtual light barrier is crossed can be small and it tends to differ for every pair of lamps. Moreover, in contrast to conventional light barriers, the illuminating signals are more diffuse and the field of view of the sensor is wider to cover a larger area of interest. In addition, multi-path can exist. This means that signals are often only partially blocked when the barrier is crossed.

To address this challenge, EyeLight employs a delta technique. For a given transmitter-receiver link, it measures the delta change in received signal power when the ON-OFF transition occurs and compares it with a delta obtained under reference conditions (i.e., an occupied room). The signal power delta effectively captures how much light from the signaling transmitter is reaching the sensor. It subtracts out all light from other sources, assuming it remains constant over the duration of one slot. If the measured delta significantly deviates from the reference delta, it means that a change between the transmitter and receiver has occurred.

More precisely, let $P^{j_k}_{i,ON}$ and $P^{j_k}_{i,OFF}$ denote the mean power measured by the k-th sensor on node j while node i is in the ON and OFF phase of its signaling, repectively. We define the delta as $\Delta^{j_k}_i = P^{j_k}_{i,ON} - P^{j_k}_{i,OFF}$.

Note that both the terms effectively sum all light power reached at the sensor k, including both the power from ambient light (natural light and illumination from lamps other than i) and signaling light power received from lamp i. That is $P^{j_k}_i = P^{j_k}_{ambient} + P^{j_k}_{i,received}$. During OFF phase of lamp i, $P^{j_k}_{i,received}$ becomes zero and assuming no change in ambient lighting between ON and OFF phases, it follows that $\Delta^{j_k}_i = P^{j_k}_{i,received,ON}$. This means *the delta value is effectively the light power reflected from node i to sensor j_k during the ON phase of node i*. When a person crosses the link between node i and j, the person can either block light or reflect more light from node i to receiver j_k, depending on the exact position and the reflectivity of the person's hair, skin and clothes. In either case, that causes $P^{j_k}_{i,received,ON}$,

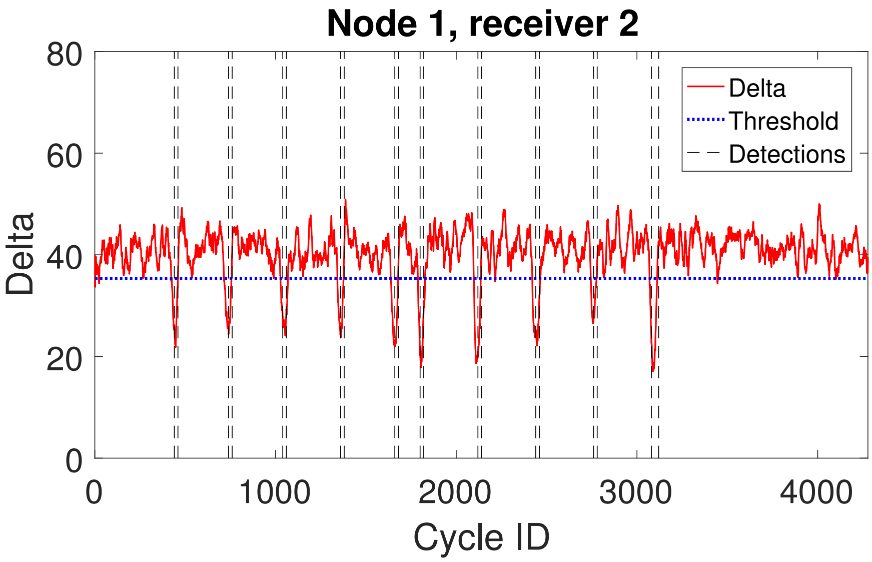


Figure 2.4: Virtual light barrier crossing detection.

and in effect $\Delta_i^{j_k}$, to deviate from the normal level.

This observation becomes the key for our light barrier crossing detection method called *Delta algorithm* (Algorithm 1). Going back to the example of receiver j_k, in each cycle, we calculate the term $\Delta_i^{j_k}$ as described above, then check if this term exceeds a preset threshold range. To reduce noise on the series of calculated delta values, we first apply Hampel filtering to remove outliers and then a low pass filter to smooth the signal. The algorithm then uses a windowing approach (set to 1s) and outputs a detection when the majority of delta values in the window exceed the threshold. We set the threshold based on the mean and the standard deviation of the delta values in the baseline dataset (when the room is unoccupied). (For our prototype, we empirically choose threshold to be $baseDelta \pm 2 * baseStd$). Fig. 2.4 illustrates one output example of the delta detection algorithm, where receiver 2 on node 1 points to node 2's direction, and a person passes 10 times the light barrier between node 1 and 2.

Given detections from either the Spike algorithm and Delta algorithm, we seek to infer the location of the person. For *Spike algorithm*, based on detections of a user or her shadow in the field of view of different receivers, EyeLight derives the user location based on the positions that these receivers are pointing to. We assign a weight for each receiver based on the magnitude of the deviation of the received light power from the baseline level. The

Algorithm 1 Delta algorithm - light barrier crossing detection

Input: $readings$ from node j_k, $baseDeltas$, $baseSTD$
Output: $events$
while *next cycle exists* **do**
  $cycle$ = getNextCycle()
  for $i = 1 \rightarrow numOfNodes$ **do**
    $P^{j_k}_{i,ON}$ = mean(*cycle period during ON phase of node* i)
    $P^{j_k}_{i,OFF}$ = mean(*cycle period during OFF phase of node* i)
    $\Delta^{j_k}_i = P^{j_k}_{i,ON} - P^{j_k}_{i,OFF}$
    Update $W^{j_k}_i$ *- running series of* $\Delta^{j_k}_i$
    $hampelfilter(W^{j_k}_i)$
    $lowpassfilter(W^{j_k}_i)$
    if $|\Delta^{j_k}_i - baseDeltas^{j_k}_i| > 2 * baseSTD^{j_k}_i$ **then**
      increase count($events^{j_k}_i$)
    if *end of 1-sec window* **then**
      if (count($events^{j_k}_i$) > window / 2) **then**
        $detection^{j_k}_i = True$
      count($events^{j_k}_i$) = 0

final location of the user is estimated as the weighted average of the locations to which the receivers are pointing to. For the *Delta algorithm*, we estimate the location of the user to be the center point between the transmitter and the location the receiver is pointing to.

Note that *Spike algorithm* and *Delta algorithm* compliment each other. The Spike algorithm provides better coverage (any movement in an area surrounding the receiver would be detected) but its location estimation is coarse-grained. In contrast, the Delta algorithm easily pinpoints which transmitter-receiver link the person crosses, but it loses track of a person that does not cross a light barrier link. To obtain both large coverage and fine-grained localization, one can combine the results from both algorithms, for example, by calculating the centroid of their estimated locations.

2.5 Room Activity and Occupancy Recognition

In this section, we introduce the room activity recognition and occupancy classification module. The study focuses on a conference room, with activities and occupancy levels categorized as in Table 2.1. This module uses a supervised machine learning approach based on a feature vector of light power measurements. For other types of rooms, our activity classifier needs to be trained separately to classify different set of activities that commonly happen in these rooms.

The features to be used have to cover all the room's different activity spots, thanks to the non-LOS nature of shadow based tracking. Based on our hypothesis, detecting the room's occupancy and different activities can be inferred from the sources of movements and light settings at different locations. For example, during a presentation activity, the ambient light is usually dimmed and most light received is coming from the projector. One can think of using the delta values and base light level readings during OFF phase of the transmitter for the feature vectors. However, limiting the features to only these two values might cause losing information needed for the classification. Also, the effectiveness of these features depends directly on the base light level, that may change from time to time. Therefore, to capture the temporal and spatial variability of light settings, we use the readings from all the receivers in the room; values for each receiver are 12 average readings of 6 timeslots (including ON and OFF phases). We include the readings from all the time slots since this enables our system to distinguish the source of the lights from multiple directions. The readings are averaged over a span of time window w. We choose w to be long enough to capture the different activities and movements by users indoors. Since humans walk on average 1.4 m/s [51], we vary this time window from a second to a minute long. We only report the time window that maximizes the classification accuracy.

Our activity and count recognition approaches uses ensemble learning, specifically AdaBoost.M2 [52]. In Adaboost, the classification results of other learning algorithms ('weak learners') are combined into a weighted sum that represents the final output of the boosted classifier. AdaBoost is able to tweak adaptively the weak learners without prior knowledge about their performance. We use regularized linear discriminant analysis (LDA) learners as weak learners. We train the room activity and occupancy ensemble classifiers with the feature vectors labeled with the activity index and occupancy category label, respectively.

Table 2.1: Room activity and occupancy categories

Activity		**Occupancy**	
Index	Room Activity	Human Count	Category
0	Empty Room	0	Empty Room
1	Sitting at/near Table	1	Single Person
2	Whiteboard Discussion	2-3	Few People
3	Projector Presentation	> 3	Many People
4	Single Person Rehearsing		
5	Conducting Experiments		

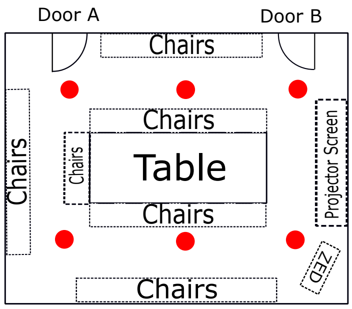
2.6 EyeLight prototype and testbed

In our prototype, we use an off-the-shelf Ecosmart 65W BR30 LED bulb as the transmitter for each light node. This light bulb contains an AC-to-DC module to provide DC power source to a series of LED chips. For our experiments, we remove this AC-to-DC module and feed 40V DC source directly from a DC power supply to the LED chips. We use a microcontroller (MSP432) to control a power MOSFET (IRFL520) as a switch to drive much larger current needed for the LED lamp. For *timeslot assignment*, to support 6 nodes, we divide each cycle (8ms) into 6 even timeslots.

In the transimpedance amplifier, we use the LF356 op-amp [53] which has low input noise voltage (12nV/$\sqrt{Hz}$) and suitable for photosensor amplifier task. The feedback resistor is 10MΩ to maximize the transimpedance gain. In the later stage, further amplification is achieved by using INA126 [54], an instrumentation amplifier with low noise characteristics.

We use TI MSP432 Launchpad [55] to control both transmitter and receiver operations. The MSP432 Launchpad also offloads data measurement through Wi-Fi to our processing server with the help of a TI CC3100 BoosterPack [56].

We built 6 light nodes and placed them inside a conference room (size $7.5\times6m^2$, ceiling height $2.74m$), as shown in Fig. 2.5. All circuit components for each light node , including the microcontroller (MSP432 Launchpad), receiver boards, synchronization board, power board, were placed on a woodplank together with the LED light bulb. We placed 4 receivers around each LED bulb, pointing to different directions; each photodiode is titled $\theta = 10^0$ compared to the vertical line. This placement of photodiodes increases the number of virtual light barriers in the room to detect human presence. To construct groundtruth, we placed



(a) Room topology.

(b) Conference room.

Figure 2.5: EyeLight testbed. There are 6 light nodes with distance between adjacent pair is 2.5m. The room has a central table, a number of chairs, and a projector screen.

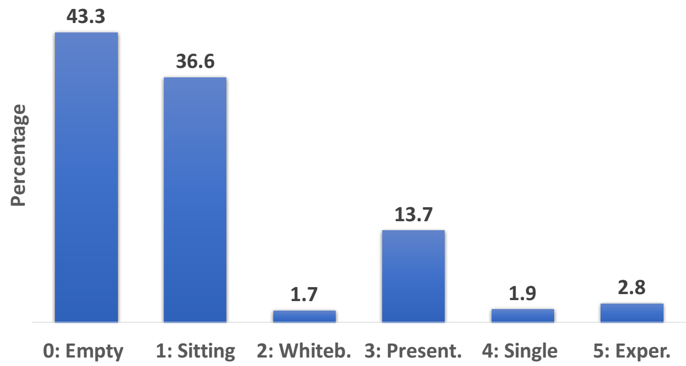
a ZED depth camera [57] in the corner of the room. The camera records videos of the room with depth information, and these videos are later manually processed to rebuild the positions of all persons inside the room.

2.7 EyeLight evaluation

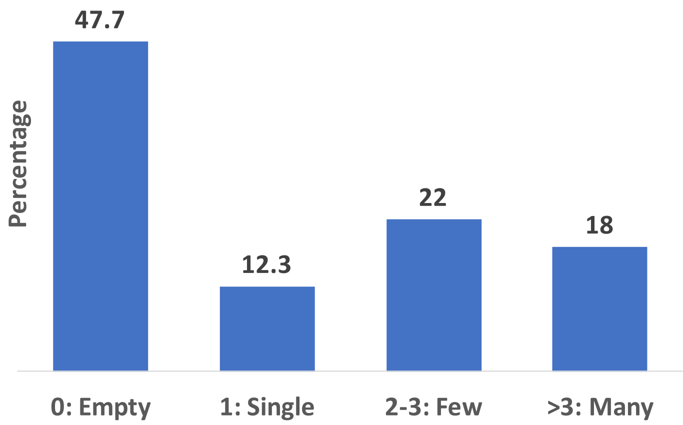
We collected data using our testbed in a conference room for 5 days over multiple weeks. For each day, we recorded data during normal working hours, the total number of hours recorded being 28.5 hours. Different users entered the room, including visitors, staff, faculty and students. Different lighting settings and different chairs organizations have been conducted during these days. We collected the base light level for the Spike and Delta algorithms at the beginning of each day.

2.7.1 Light barrier crossing detection accuracy

The output of the Delta detection algorithm for each photoreceiver is a binary detection: for each second, whether there is shadow casted by the adjacent node on the floor where the receiver is looking at. To evaluate the accuracy of our Delta detection algorithm, we conduct an experiment in which several test subjects walk in the room across all the lamps. Fig. 2.7 shows the True Positive Rate (TPR) and False Positive Rate (FPR) of the delta detection algorithm for different photoreceivers. TPR is the ratio of the correctly detected events over the total number of proximity events, and FPR is the ratio of the incorrectly



(a) Activity Identification



(b) Occupancy Estimation.

Figure 2.6: The distribution of different activities and occupancy categories in the dataset.

detected events over the total number of testing cases when no person is in the vicnity of a sensor. The receivers in the figure are the ones pointing to an adjacent light bulb. The average TPR across all receivers is 82.17% and the average FPR is 5.77%. Among all receivers, only receiver 5.3 has low TPR (6%). Given our conference room has dark carpet with low reflected light, the TPR and FPR value reported here are reasonably good. Also, this is the performance for each single receiver; we expect that by combining multiple receivers together, the accuracy of the whole system would be higher.

2.7.2 Localization error

Fig. 2.8 shows the localization error for single-person tracking scenarios, using three different methods: using only spikes detection, using only delta detection, and combined. Delta

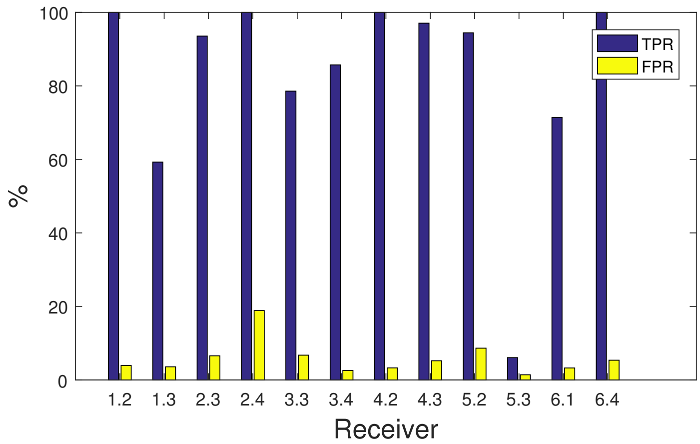


Figure 2.7: True positive rate and false positive rate of delta detection algorithm for all different sensors.

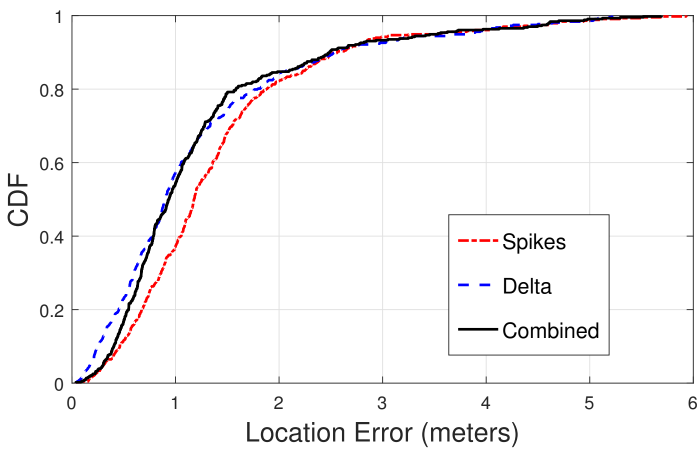


Figure 2.8: CDF of localization error for three cases: 1) using only spikes detection, 2) using only delta detection, and 3) combined detection.

detection shows lower localization error (median 0.89m and 90 percentile of 2.5m) than spikes detection (median 1.18m and 90 percentile of 2.56m). However, the spikes detection is achieving this localization error while covering 94% of the time in which the user is inside the room compared to 69% for the delta detection. It is clear that there is a tradeoff here between the coverage and localization accuracy. Therefore, we also propose the combined version of the two algorithms, which achieves a 0.94m median error and better coverage rate than the delta detection.

Figure 2.9: Confusion matrix for activity identification in a conference room. Each value in the matrix is rounded to two decimal places. Total size of the dataset is 102889 feature vectors, corresponding to 28.5 hours.

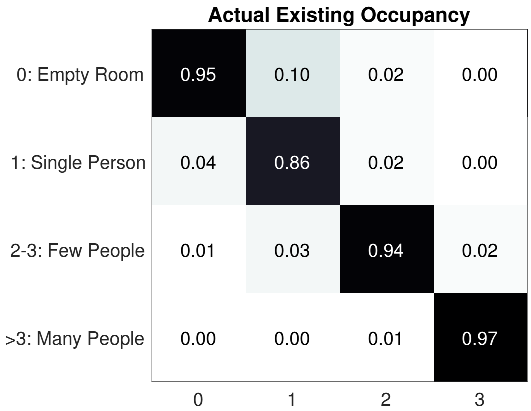


Figure 2.10: Confusion matrix for occupancy estimation in a conference room. Each value in the matrix is rounded to two decimal places. Total size of the dataset is 1710 feature vectors, corresponding to 28.5 hours.

2.7.3 Room Activity Recognition and Occupancy Estimation

We evaluate our room activity recognition classifier by 10-fold cross validation over the whole collected dataset using random partitioning. Each feature vector represents the average readings over a 5-second period, which maximizes the classification accuracy. Fig. 2.9 shows the confusion matrix for the classification results of our activity recognition classifier. Each column represents the actual activity performed by the user and each row shows the activity as classified by our system. Therefore, each column should add up to one before rounding, but each row should not add up to one. The overall classification accuracy is

93.78%; however, if we break down the TPR for each activity, we can see the performance degrades for categories 2: whiteboard discussion, 4: single rehearsal and 5: conducting experiments. These activities represent a small fraction of the collected data as presented in Fig. 2.6(a), and therefore, the classifier likely has not enough data to accurately capture the true model of these classes. Also, class 3, presentation in the dark, is easily misidentified as class 0, empty room, since the room is almost dark, and during presentation there are not many movements to capture. However, we expect collecting more data specially for these classes will decrease the classification error.

EyeLight is able to distinguish 4 classes of occupancy of a room, by classifying the readings coming from all the nodes inside. Each feature vector represents the average readings over a 10-second period, which maximizes the classification accuracy. We used the same evaluation procedure of the activity recognition classifier (10-fold cross validation). Fig. 2.10 shows the confusion matrix for occupancy estimation classifier. Each column represents the actual existing occupancy in a conference room and each row shows the occupancy category as classified by our system. The overall accuracy of the classifier is 93.7%, while the TPR for single person class is lowest among all the classes with 86%. A single person staying in a conference room is not a common event, so the dataset for this class is not enough. Detecting a single person is thus more challenging than multiple persons specifically, since the collected data for single-person class is also the lowest among the four classes as in Fig. 2.6(b). Moreover, a single person induces low effect on the light especially when not moving (e.g., sitting near the table and working with on a laptop). Therefore, we moved from five-second feature vectors, as in the activity recognition, to a ten-second feature vector in order to capture more of these rare movements for a single user. Again, we expect that collecting more data for this class would improve the classification accuracy.

2.7.4 Microbenchmark experiments

Distance between nodes. We increase the distance between the two nodes and measure the delta values received in one receiver for each distance between two nodes (Fig. 2.11(a)). As the distance between two nodes increases, the delta value becomes smaller; starting from 3.5 meters, this delta is too small to distinguish from noise, rendering EyeLight ineffective

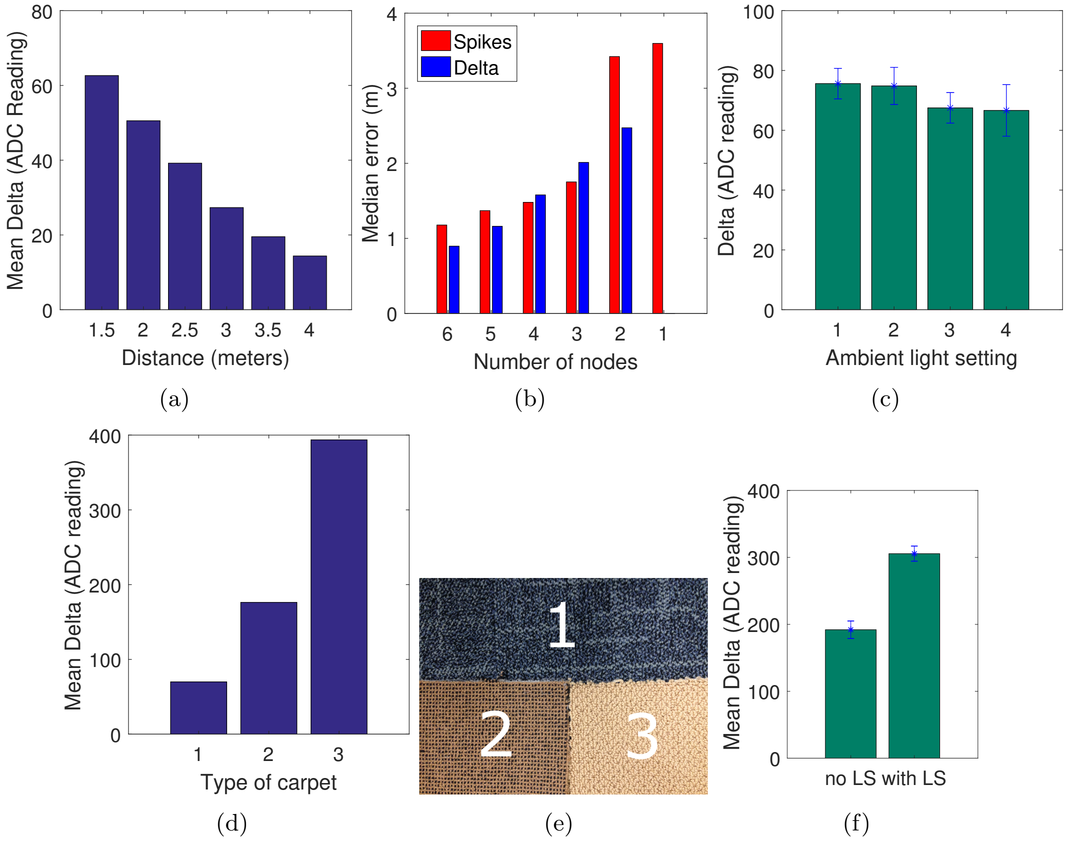


Figure 2.11: (a) Delta values for different distance between two nodes. (b) Location median error for different number of nodes. (c) Delta values for different ambient light settings. (d) Delta values for different types of floor carpets.(e) Types of carpet in (d). (f) Effect of lamp shade.

to use.

Number of nodes. We measure the localization median error of two algorithms (Spikes and Delta) with reducing number of nodes to cover our conference room (Fig. 2.11(b)). Note that there is no data for single node case of the Delta algorithm detection, since it needs at least a communication link between two nodes. As expected, the location accuracy reduces as the number of nodes decreases, because either the number of guarded locations (for Spike detection) or the number of virtual light barriers (for Delta detection) decreases.

Ambient light. We test different ambient light settings in our conference room: no ambient light, only ceiling lights turned on, only side lights turned on, both ceiling lights and side lights turned on. The mean and standard deviation of delta values for each light setting over a period of time is shown in Fig. 2.11(c). For each ambient light setting, the standard deviation is small, allows the delta algorithm to work efficiently. However, the mean value of deltas slightly differs between light settings, suggesting that the system might need to calibrate for several times a day, when the ambient light setting is changed.

Different types of carpets. Another factor that affects the efficiency of the delta-based virtual light barrier crossing detector is the reflectivity of the floor carpet materials. The carpet inside our conference room is dark, and thus reflects less light. To see the applicability of our detection algorithm on other types of carpets, we tested a light node facing different types of carpets (Fig. 2.11(e)) and compute the delta values (Fig. 2.11(d)). As can be seen, two other carpets have brighter surface, giving much larger delta values. Therefore, we believe EyeLight is also suitable to work with other room carpet, with even better performance. For other floor types, such as tiles, wood, due smoother surface, they reflect light even better than carpets, thus are also applicable in EyeLight.

Lamp shade. In all previous experiments, we tested with commercial light bulbs without lamp shades. To show the effect of lamp shade on the transmitted signal, we compared the average delta values for lamps with and without lampshade (Fig. 2.11(f)). The result shows that with lamp shade on, the average delta value actually increases. One might think that lampshade would reduce the intensity of the light from the transmitter, weakening received light power at the receiver. In fact, however, the lamp shade distributes light more evenly over the floor area under the lamp, thus improve the sensitivity of the receivers looking at different spots on the floor.

2.8 Discussion and Conclusion

We proposed a device-free indoor tracking, occupancy estimation and activity recognition system that can be integrated in light-bulbs. The key idea is to create a mesh of reflective virtual light barriers across networked light bulbs to detect occupant movement. We found that our high-sensitivity photo-sensing circuit can detect minute light changes (shadows) even on dark carpeting, and that a time division pulse signaling scheme allows differentiating the light nodes causing shadows on the floor. With our 45 m^2 conference room prototype system with 6 light bulbs each carrying 4 receivers, we further found that the sensing system can achieve a 0.89m median localization error as well as 93.7% and 93.78% occupancy and activity classification accuracy, respectively.

Our current system still has several limitations that could be addressed in future work. *First*, EyeLight requires more than one lamp per room for fine-grained user tracking. Fortunately, the small size of LED lights makes it easier to add additional lights in rooms. *Second*, EyeLight so far focuses on tracking a single person per room. It could track multiple persons as long as they cross different virtual light barriers, while multiple persons walking together leads to mixed shadows. *Third*, EyeLight needs to adapt to different light settings, such as different times of the day, rooms with outdoor light passing through windows. Currently our prototype works in a conference room without windows, where measured illuminance of light reflected from the floor is under 5 lux. In a room with outdoor light entering through windows, the current receivers saturate. However, techniques like Adaptive Gain Control, as used in other systems dealing with high dynamic range, can be added to EyeLight to improve its robustness. An adaptive system is also needed to keep track of the change of the baseline light level. We leave such designs for future work.

Chapter 3

Verification: Accuracy Evaluation of WiFi Fine Time Measurements on an Open Platform

3.1 Introduction

WiFi positioning has established itself as a key positioning technology that together with GPS is widely used by mobile devices. Yet, precise indoor positioning, to obtain room-level location or to support indoor navigation has remained stubbornly challenging due to the large effort required to create and maintain WiFi surveys. Similar challenges exist in urban canyons, where GPS accuracy is degraded due to multipath and WiFi positioning cannot entirely replace it.

Among many other solutions, researchers have long advocated for RF time-of-flight positioning methods. Time-based WiFi ranging techniques have been introduced in [58, 59] and followed by several improvements [60–64]. More recently, IEEE 802.11-2016 standardized [65] a Fine Time Measurement (FTM) protocol that supports such ideas. According to the WiFi alliance, this ranging system offer meter-level accuracy [66]. Major WiFi chipset vendors have already released WiFi chipsets that support FTM protocol based on 802.11REVmc and Android P support has been announced [67]. Beside the 802.11 standard documents, there are few details about implementation techniques and the performance of such ranging systems [68] using the off-the-shelf WiFi chipsets.

Given the momentum building around this technology, this work therefore sets out to verify the research and standard accuracy claims. We describe an open tool for the research community to experiment with WiFi fine-time measurements, based on the the Intel Dual Band Wireless-AC 8260 and 8265 cards and their open-source Linux driver. We also develop a systematic methodology for measuring the performance of time-based ranging systems to enhance the repeatability of such experiments by gradually introducing additional reflectors in the environment that add multipath propagation.

Generally, WiFi time-of-flight ranging estimates distance by measuring the round-trip time of a signal between a station and an access point. It promises several advantages. First, time-of-flight is linearly dependent on range (as opposed to received signal strength, for example), which should allow a ranging error nearly independent of distance. Second, the timing of the leading edge of a signal is less dependent on multipath than the signal amplitude.

Given this, one might expect that the accuracy of time-based ranging systems that have nanosecond resolution should not be affected by multipath while the line-of-sight (LoS) transmission is not blocked. We found that the accuracy of this ranging system at 2.4 GHz and up to 40 MHz bandwidth is significantly affected by the non-line-of-sight (NLoS) components of the transmitted signal in non-open space environments. At 5 GHz and 80 MHz bandwidth the results approach meter-level accuracy in the indoor LoS environment but become unreliable in a NLoS environment at distances above 20m. It therefore, requires denser access point deployments to realize these gains.

Given that speed-of-light signals cannot arrive too early, one might also expect that ranging errors are biased towards long estimates rather than short estimates. We found that some configurations of the system, in particular 2.4 GHz with 20-40 MHz bandwidth at the access point, frequently outputs short estimates and without calibration can produce negative estimated ranging distances, when the ground truth distance is less than 6 meters. This bias occurs even when the LoS component of the signal is dominant to the NLoS components. Using external antennas with different orientations can help to alleviate this problem. Also, using longer cables to connect the two antennas to the card (which increases the propagation delay) can correct this, along with filtering out measurement noise. After canceling out the offset, the estimated distances are within (±1) m of the actual distances in an outdoor open-space setup.

In summary, the major contributions of this work are as follows:

- Conducting an evaluation and verification of the performance claims of WiFi time-of-flight ranging and positioning in several benchmark and real-world environments.

- Introducing, analyzing, and calibrating an open platform for conducting WiFi time-of-flight experiments to the research community.
- Proposing a repeatable measurement framework for evaluating time-based ranging systems.
- Confirming the expected meter-level ranging accuracy in open-space outdoor environments, while showing that multipath environments remain a challenge at least at lower bandwidths.

3.2 Background

A wireless ranging system estimates the distance between two devices by sending a wireless signal between them. As the signal travels between the two devices, its properties change over the distance. These properties include amplitude, frequency and phase. Moreover, given the propagation speed of the signal, a ranging system can estimate the distance by estimating the time the signal takes to travel between two devices. In this section, we illustrate time-based ranging systems and their challenges.

3.2.1 Multipath Challenge

A key challenge that limits ranging systems is multipath, where the transmitted signal is reflected in the environment and reaches the receiver's antenna by more than one path. These paths have different lengths resulting in different RTTs. Therefore, ranging accuracy depends directly on whether the RTT measurement is based on the direct path or a reflected path. The main challenge, in presence of multipath, is how to distinguish the direct path signal from the reflected signals. In case of non-blocked direct path, this problem seems to be solvable for time-based ranging, by simply picking the first received signal. However, dealing with signals that travel with speed of light complicates the problem.

Bandwidth and raw localization resolution. Detecting the arrival of a packet is challenging since a difference of 1 ns could result in an error of 1 foot for the RF ranging systems (speed of light $\approx$ 1 foot/nanosecond). Therefore, a fine resolution clock with 1 ns or higher is needed for 1 foot raw accuracy. Another factor that limits the accuracy

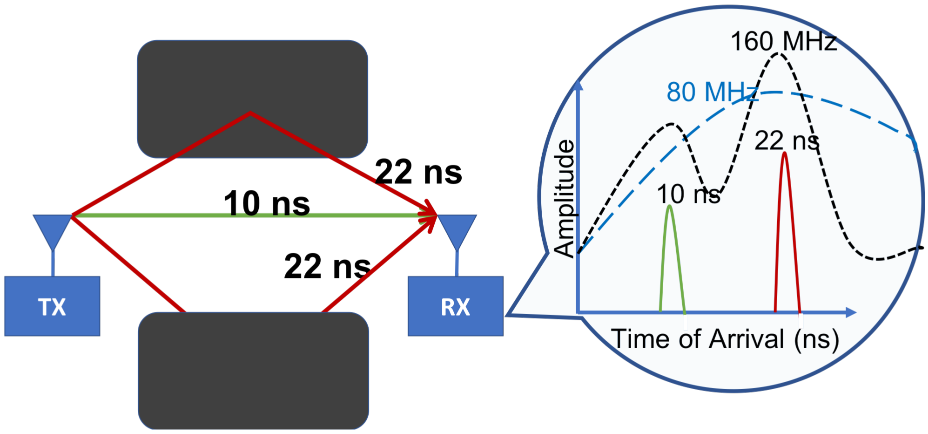


Figure 3.1: Multipath problem.

resolution for packet detection algorithms is the channel bandwidth. For example, WiFi signal is sampled once every 50 nanoseconds for a 20 MHz channel, during this period, the signal travels 15 meters. Therefore, distinguishing between two signal spaced by a distance less than that raw resolution is challenging problem. Prior super-resolution spectral signal processing techniques [69, 70] can improve this raw resolution. Fig. 3.1 shows the multipath problem, and illustrates the channel bandwidth effect on the ranging error. In this figure, the transmitted signal reaches the receiver through three main paths, line of sight (LoS), and two reflections from parallel planes (same material and same distance to transmitter). However, the LoS component of the signal reaches the receiver first, the NLoS components arrive afterwards with the same signal phases, resulting stronger received signal through constructive interference. With enough bandwidth, hence higher ADC sampling rate, the receiver will be able to sample enough to distinguish between the first arrival through direct path and the multipath reception.

3.2.2 Evaluation Challenges

Different environments lead to different multipath profiles, hence result in different performances for ranging systems. Therefore, it is challenging to produce repeatable and generalized results to evaluate a ranging system . Moreover, details about signal processing algorithms implemented on current off-the-shelf cards that support the FTM ranging system are not available, even for the open source drivers. These physical layer algorithms are implemented in the firmware of these cards. As a result, there is no information about how the packet arrival is being detected, how the implementation deals with multipath problem, or what is the bandwidth used for the packet arrival detection. Even details about the clock resolution of these cards are not available. In this work, we present a measurement framework for evaluating the FTM ranging system, even without knowing beforehand the answers of the previously mentioned questions.

3.2.3 Fine Time Measurement

IEEE 802.11-2016 standardized a Fine Time Measurement (FTM) protocol that enables a pair of WiFi cards to estimate distance between them. Fig. 3.2 illustrates the details of the FTM protocol. An initiator is a station (STA) that initiates the FTM process by sending a FTM Request to a corresponding access point (AP). An AP that supports the FTM procedure as a responding device (Fig. 3.2) is called a responder. Based on the AP response, the protocol agrees or refuses to continue the ranging process. In the case of agreement, the AP/responder starts to send FTM message and wait for its ACK. The RTT is estimated based on the transmission timestamp of the FTM message and the reception timestamp of its ACK. The AP may send multiple FTM messages, but have to wait for acknowledgement, before sending a new message. Fig. 3.2 shows an example of one burst with 3 FTM messages, with ASAP mode set to 1[1]. The RTT is calculated for n FTM messages as follows:

$$RTT = \frac{1}{n}(\sum_{k=1}^{n} t_4(k) - \sum_{k=1}^{n} t_1(k)) - \frac{1}{n}(\sum_{k=1}^{n} t_3(k) - \sum_{k=1}^{n} t_2(k))$$

Generally, the protocol excludes the processing time on the STA by subtracting it $(t_3 - t_2)$ from the total round trip time $(t_4 - t_1)$, which represents the time from the moment the FTM message is being sent (t_1) to the moment the ACK is being received (t_4). This calculation is repeated for each FTM-ACK exchange and the final RTT is the average over the number of messages in the burst.

[1]STA is ready to receive FTM messages and hence, capable of capturing timestamps associated with an initial FTM message and sending them in the following message

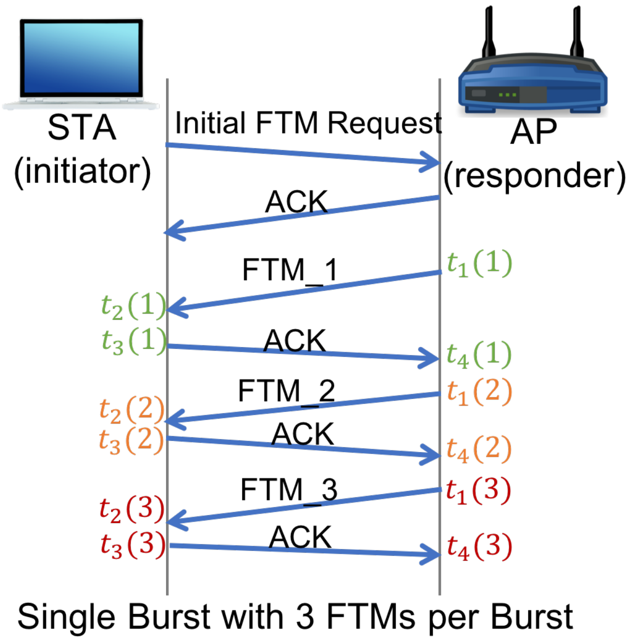


Figure 3.2: FTM Protocol Overview.

3.3 Open WiFi Time-of-Flight Platform and Basic Ranging Calibration

3.3.1 Open WiFi FTM Tool

Hardware. While several vendors offer WiFi implementations with support for the FTM standard, we chose the Intel Dual Band Wireless-AC 8260 and 8265 since they are the only ones for which we were able to obtain open software support for accessing FTM measurements (an open source Linux driver with an experimental FTM implementation). In practice, we found that this driver requires further modifications to be usable for FTM measurements. We refer to these cards as WiFi card A and use them as the station to take measurements as well as the access point in some experiments. In access point mode, these cards unfortunately only support the 2.4 GHz band and 20/40MHz channels but they do support 5GHz and higher bandwidth as clients. Moreover, we found that the ASUS Wireless-AC1300 RT-ACRH13 APs is configured to respond to FTM requests out of the box. This AP uses the Qualcomm IPQ4018 chipset and also supports the 5GHz band with higher bandwidths. We refer to this AP as AP B.

Software versions for open station. In our framework, APs and STAs use Linux

AP label	20 MHz (2.4 GHz)	40 MHz (2.4 GHz)	20 MHz (5 GHz)	40 MHz (5 GHz)	80 MHz (5 GHz)
WiFi card A	-6.8 m	-6.8 m	Not available	Not available	Not available
AP B	Not accurate (-1000 m)	Not accurate (-998 m)	-15 m	-5 m	1.8 m

Table 3.1: Comparing the average ranging accuracy for using different WiFi chipsets as the AP, while fixing the STA using WiFi card A, under different bandwidth and band configurations. The average ranging accuracy is reported in meters for 1 meter actual distance.

kernel version 3.19.0-61-lowlatency. Although, the FTM protocol is implemented in newer kernel versions, it is only supported by the backport LinuxCore releases of the IWLWIFI driver [71]. We therefore use the IWLWIFI driver from the LinuxCore30 release, along with firmware version 31. The station can be configured with the iw (nl80211 based CLI configuration utility for wireless devices in Linux) Linux command line tool.

Configuring WiFi cards as FTM responding access points. The node can be configured as access point using hostapd (we used version 2.6). Configuring a node as AP through hostapd does not automatically enable it to respond to FTM protocol messages. We therefore modified the IWLWIFI driver to activate the responding feature when it is configured as an access point and make this patch available. Through hostapd configuration, the AP can publish their FTM support as a responder through the beacon frames.

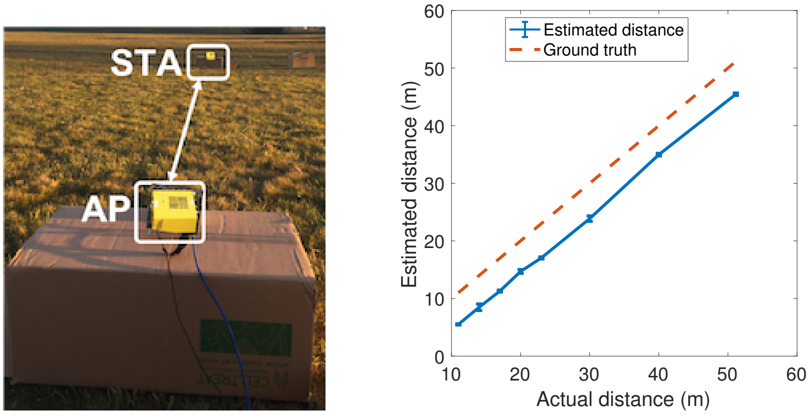
Initiating FTM requests. There are two options for triggering an FTM ranging request at the station. The first option is to leverage Linux Debugfs, a filesystem that enables communication between kernel and user space. The second option, which we adopt in this work, is to use iw command line tool, along with a patch [72] that adds the FTM feature to the iw command and enables the STA to initiate the FTM process by sending FTM request. An initiating STA needs to acquire specific information about the AP in order to send the FTM request. This information includes MAC address, supported bandwidth, and frequency. Therefore, our tool starts the process by scanning the surrounding APs, in order to acquire the needed information. If a STA sends a FTM request to an AP, that doesn't support FTM, this AP will not respond and the STA has to wait for timeout to return unsuccessful ranging status. To avoid this delay, our tool send FTM requests only to the APs that supports FTM protocol. According to the standard [65], each AP that supports FTM as a responder shall publish this information in the beacon frames, (a specific bit in the extended capabilities record refers to FTM responder support.

RTT calculation. After initiating the FTM process, the AP starts to send FTM frame automatically and waits for its ACK to estimate the RTT. This process is implemented in the proprietary firmware but based on the standard we know that in order to remove the processing time for the initiating STA from the RTT, the responding AP transfers the timestamp values it captured (t_1 and t_4) to the initiating STA in the follow up FTM frame. The initiating STA is the responsible for computing the RTT, this computation is done in the firmware. By increasing the number of samples per burst, the AP sends several FTM frames in a sequence and the initiating STA estimates the RTT for each pair of FTM message/ACK. The RTT for each pair of FTM message/ACK is not available in the driver, only the averaged RTT over the burst in picoseconds along with the corresponding distance in centimeters and average received signal (RSSI) are finally returned.

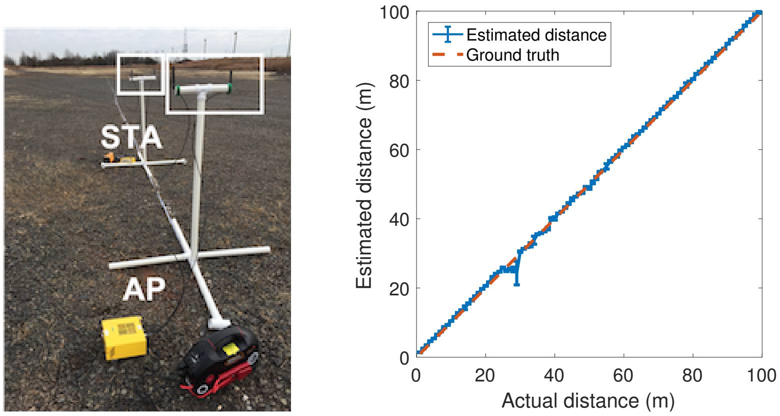
Tool limitations. Extracting information including CSI, phase and measurement per antenna is currently not available due to firmware limitations. Similarly, ranging accuracy depends on the communication bandwidth which implies having an analogue-to-digital converter (ADC) that can sample at that rate. WiFi card A can be configured as access point for up to 40MHz and should support up to 160MHz bandwidth as a station with an appropriate 802.11mc compliant access point. So far we were able to confirm support up to 80MHz.

3.3.2 Experimental Setup

Our experimental setup consists of two small form factor PCs (containing WiFi card A), one of them configured as AP and the other one as STA. We refer to this setup as WiFi card A setup. In a second setup, we use AP B, while still using the same small form factor PC as station. We refer to this setup as AP B setup. In these two main setups, we evaluate the FTM protocol supported by WiFi card A and AP B. These WiFi chipsets require two antennas. We use omni-directional antennas with 6 dBi gain. To extend the height of the antennas, we use 6-feet CNT-240 cables, in which the velocity of the signal is 83% the speed of light in a vacuum and the attenuation for 2.4GHz is 12.9 dB/100 feet. Along with the cables, we use PVC pipes to fix the antennas on specific height while avoiding disturbing



(a) Grass ground (0.7 m height).



(b) Rocks ground (1.4 m height, offset corrected.)

Figure 3.3: Outdoor open space. This setup uses the WiFi card A for both STA and AP sides.

the signal. For measuring the ground truth distance, we use 400-Feet measuring tape, along with BOSCH GLM 80 laser distance and angle measurer. In experiments involving vehicles, we use GPS readings as ground truth.

3.3.3 Basic Ranging Accuracy Calibration

We start with an open space outdoor area, in which the surrounding environment is stationary. The multipath problem is minimized in this setup, in which only the ground bounce affects the measurement of the direct path. In such setup, we study two environments: 1. open green field 2. open rocks paved field.

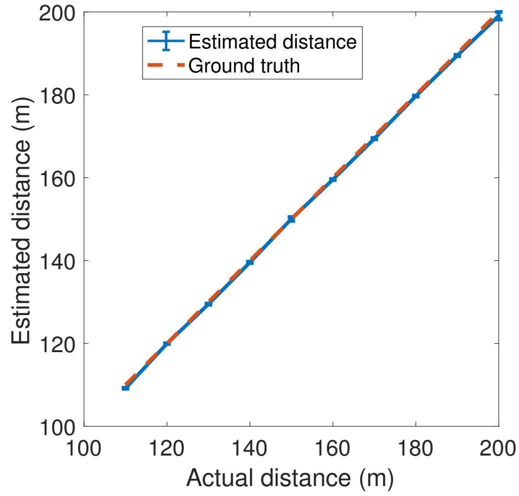


Figure 3.4: Maximum range in outdoor open space (1.4 m height, offset corrected. This setup uses the WiFi card A for both STA and AP sides.

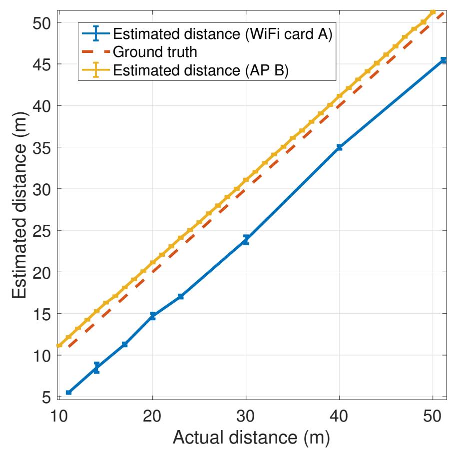


Figure 3.5: Outdoor open space grass ground (0.7 m height). Comparing ranging results for using WiFi card A as AP compared to AP B while using WiFi card A as STA in both scenarios.

Surprisingly, for the WiFi card A setup, Fig.3.3(a), and Fig.3.6(c) show that the system underestimates distance and returns negative round-trip-time estimations for short distances. Multipath effects can lead to longer paths but we are not aware of any effects that would allow the signal to arrive earlier than expected. We believe that this is due to internal calibration of the WiFi cards or multipath compensation algorithms that process the measurements in firmware before they are delivered to the driver. Our open-space stationary measurements at different distances, shown in Fig. 3.6(c), as an example, illustrate that in an open space stationary environment the mean error is constant over distance but variance increases at longer distances. The mean fixed offset is 5.7 m, which we confirmed by measuring the offset with multiple different pairs of cards.

On the other hand, ranging to AP B using WiFi card A, as in Fig.3.5 in open space, does not show underestimation of the distances compared to ranging to WiFi card A. In this new setup, STA and AP have different chipsets (belonging to different vendors), while using 80 MHz bandwidth at 5 GHz band.

Offset correction. Filtering the fixed offset can be done by either subtracting that measured offset from the readings or by cancelling this offset through the delay of long enough cables. In this work, we correct for the subtracted offset using 6 feet cables connecting the WiFi card to the two needed antennas. We use 6 feet cables, as it adds 1.83 meters on both sides resulting into 4.4 meters added delay after taking into account the speed of the signal in the used cables. This also enables us to put the antennas on reasonable height (1.4 m) and helps separate the antennas from the noise produced by the metal box containing our form factor small personal computer.

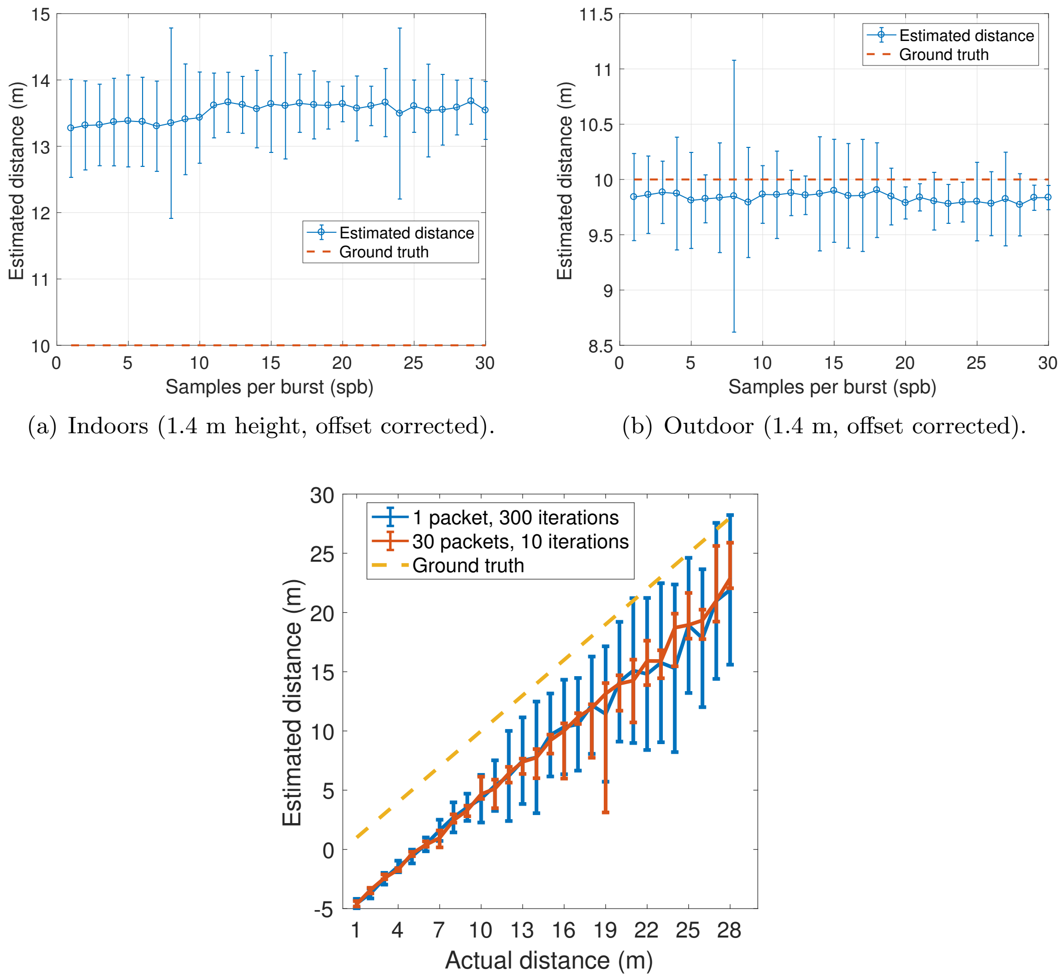
After correcting the fixed offset, Fig.3.3(b) show that in these kind of open space environment, the ranging system works accurately (within 1 meter error). However, it sometimes underestimates the actual distance, even after correcting for the offset, as shown in Fig. 3.3(b) (between 27 m and 37 m). Therefore, we shifted both the STA and AP, while preserving the same separation distance, in order to confirm that this problem is because of the environment. After shifting the STA and AP, we observed that the estimated distance returns to the normal behaviour in such open space setup. We also tested the maximum range of this ranging system in open space environment, while putting the AP and the STA on 1.4 m height. Fig.3.4 shows that this ranging system can still estimate distances up to 200 m. We did not test the ranging system for distances more than 200 m.

Bandwidth effect. Generally, increasing the signal bandwidth is expected to improve the accuracy of time-of-flight ranging systems, especially in a multipath environment, but it is unclear whether the data communication bandwidth setting affects this process. For these WiFi cards A, it is only permitted by the firmware to work as AP in 2.4 GHz band, while only passive reception is permitted in 5 GHz band. Therefore, only 20 and 40 MHz channel bandwidth settings were available to us. We experimented with both bandwidth settings and did not observe any effect on ranging accuracy.

On the other hand, AP B, that we use, supports FTM with 80 MHz bandwidth at the 5 GHz band. The results at this higher bandwidth (higher ADC sampling rate) show a significant improvement (in Fig. 3.5 and Fig. 3.12(a)), although the different band and different chipset could have also been a factor. A more detailed comparison is presented in Table 3.1. Note that in the AP B configuration with 2.4GHz, the returned ranging estimates are very unreliable which could be due to compatibility or calibration issues since the FTM feature in the open source driver used for the client is not officially supported by the chipset vendor.

3.4 Experimental Framework

In this section, we characterize the performance of the FTM ranging system through our experimental framework. We study the effect of different software and environmental parameters that affect the performance of such system. Our experimental framework consists of several experiments, indoors and outdoors to quantify the accuracy of the ranging system, and most importantly, the repeatability of these results. Therefore, we start with outdoor open space stationary environments, in which we aim to understand the basic accuracy in an ideal simplified environment (single reflection by the ground). In outdoor open-space setups, we have control on the multipath problem. For example, we can control the length of the ground bounce by changing the antenna height. We can either add second bounce or block the direct path by adding reflector parallel or perpendicular to the direct path, respectively. As these ideal environments are not common, we move after that to evaluate common and more challenging situations. For example, outdoor dynamic environments with vehicles moving and causing different kinds of reflections to the transmitted signal. Indoor setups are another example for such common environments in which the multipath problem is complicated by reflections from walls, load bearing columns, doors, and furniture.



(a) Indoors (1.4 m height, offset corrected).

(b) Outdoor (1.4 m, offset corrected).

(c) Vary distance (zero height).

Figure 3.6: Effect of varying number of samples per burst.

3.4.1 Sampling Effect

Samples per burst (spb). First, we study the effect of varying the number of FTM packets per burst, i.e. burst size, on the performance of the system. Fig. 3.6(a), and Fig. 3.6(b) show the effect of varying number of samples per burst over the estimated distance, while fixing the distance to 10 m indoors and outdoors. As the number of samples increases, the variance of the measurements decreases. Fig. 3.6(c) shows the estimated distance while varying the actual distance between the transmitter (AP) and the receiver (STA) while fixing their heights (on the ground, 0 m), in open space environment. In this ranging procedure, we change the burst size (1, and 30), while repeatedly calling the command multiple times (300, and 10). In the rest of the work, we stick with the highest

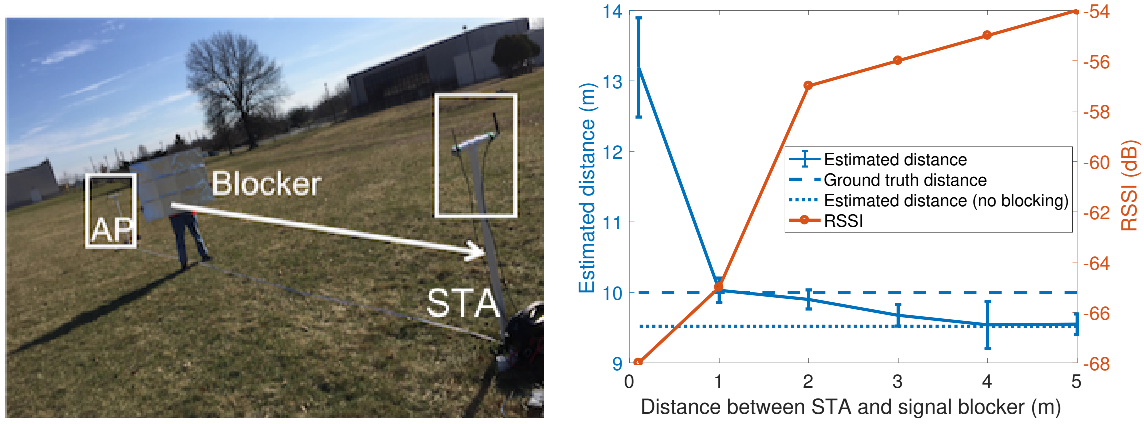


Figure 3.7: Blocking LoS reception (1.4 m height, offset corrected).

burst size (30) that minimizes the measurement noise.

3.4.2 Multipath Effect

Outdoor, blocking direct path. Even with high clock resolution and high bandwidth, ranging systems suffer from overestimating the distance while blocking the LoS reception. While this situation can practically happen, we move on to study the effect of blocking the LoS reception on the estimated distance in the same open space environment. In this setup, we use a 1.2 m $\times$ 0.9 m sheet covered with aluminum foil to block the signal between the AP and the STA. We move the blocking sheet along the line connecting the STA and the AP. Fig. 3.7 presents the effect of blocking the signal on the estimated distance along with the received signal. As the blocking sheet moves towards the STA, the estimated distance increases. The reason behind this is due to the partial blocking of the LoS. Since the 1.2 m $\times$ 0.9 m aluminum sheet cannot block the whole propagation channel in the open space, placing the blocker closer to the transmitter/receiver would essentially result in longer multipath propagation, which leads to longer distance estimation.

Outdoor, two reflectors. We start to add another reflector to the ground reflector in outdoor environment and study how the system reacts to simple multipath problem. In this setup, we fix the distance between the AP and the STA to 5 m, while aligning the line connecting the two nodes to be parallel to a side of 7-floor building. This side is 22 m long. Fig. 3.8 shows that the ranging system over the same separating distance swings between

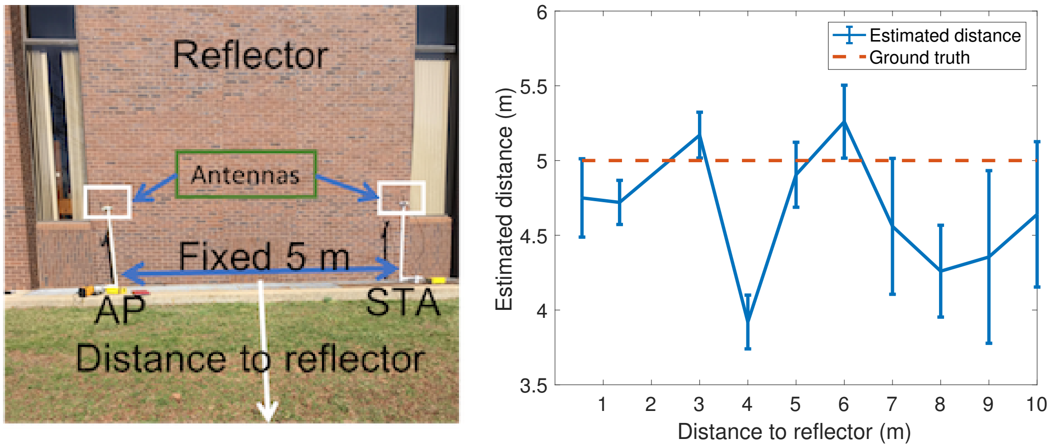
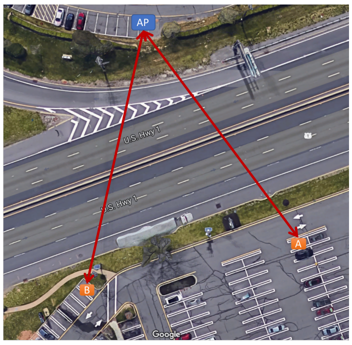
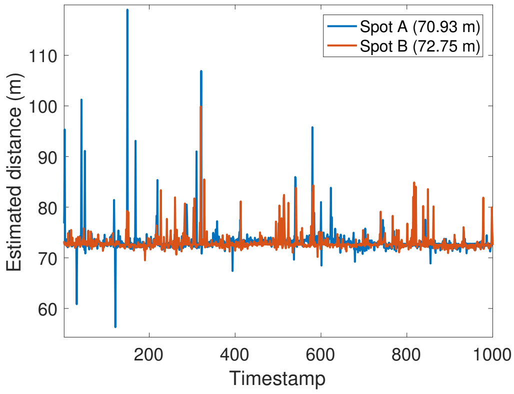


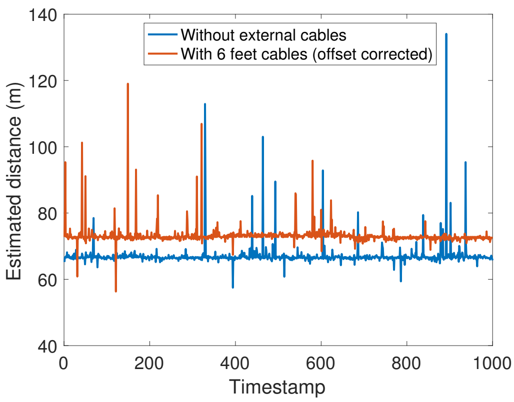
Figure 3.8: Outdoor parallel to 7-floor building (1.4 m height, offset corrected).



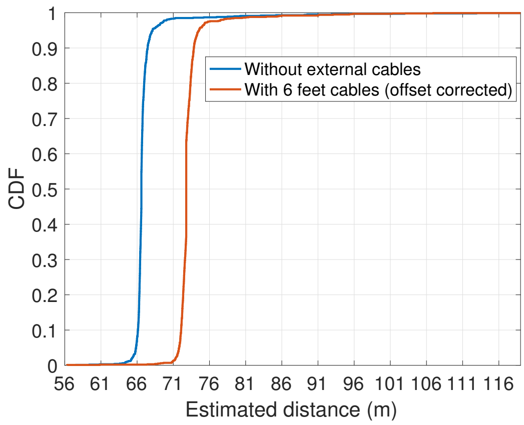
(a) Map.



(b) Temp. variations. Comparing different spots.



(c) Temp. variations. Spot A.



(d) CDF. Spot A.

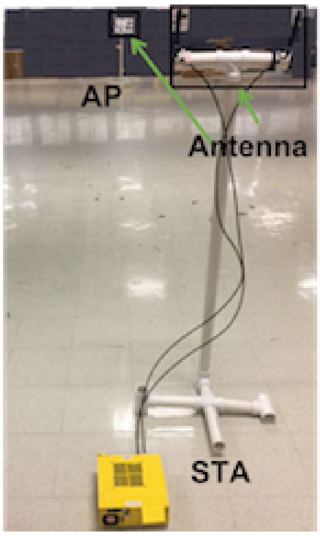
Figure 3.9: Mobile environment effect for static sender and receiver.

underestimating and overestimating the distance while varying the distance to the major reflector.

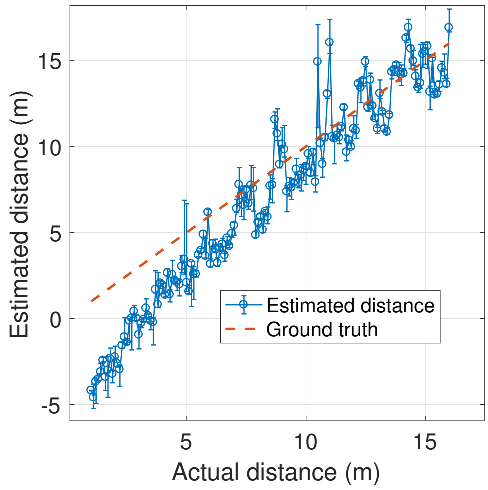
Outdoor, highway, mobile environment. In this setup, we study how a highly dynamic environment could affect the estimated distance between static transmitter and receiver. We fix the AP location, and use two different spots for the STA (Fig. 3.9(a)). We notice a medium traffic on the highway during the experiment. We use 30 packets per burst and repeat the procedure 1000 times, lasting for 6 minutes. The ground truth distances between the AP and the STA are 72.5 m, and 72.3 m, and the median estimated distances are 72.79 m (66.56 m before correction), and 72.61 m, respectively for spot A and B. The median estimated distance converges over time to the actual distance, as more samples being measured while the LoS propagation is not blocked. Moreover, this ranging system underestimates the actual distance even after correcting for the fixed offset by using 6-feet cables. On the other hand, the spikes occur when big trucks pass by and block the direct LoS propagation between the AP and the STA.

Indoor. Indoor environments are challenging for ranging systems because of the multipath problem. Few steps could produce high variant measurements, even for time-based ranging systems. In order to capture this behaviour, we conduct two indoor experiments, in which we fix the AP and the STA on the same height (0.76 m), and move the STA with a step of 10 cm (less than a wavelength of 2.4 GHz frequency). We use 30 samples per burst, while repeating this FTM measurements 10 times. These two experiments are conducted in the same single-floor building, with the ceiling height 5.3 m. Fig. 3.10(b) shows the relation between the actual and estimated distance, while varying the distance in a large experimentation room (24.2 m $\times$ 19.4 m). We repeat the same experiment in a long corridor (30.7 m $\times$ 2.5 m), as shown in Fig. 3.11(b). These experiments show that even for 10 cm step, the measurements could vary up to 5 meters in these settings. These results highlight how the multipath problem could affect such ranging systems, specially indoors. This is clearly emphasized in Fig. 3.12(b), in which we vary the distance from 10 to 10.5 m with 1 cm step. We can see that by varying with only 1 cm, the signal gets completely blocked at 10.06 m. Therefore, the user should not expect getting the same output while moving small steps, even 1 cm matters significantly.

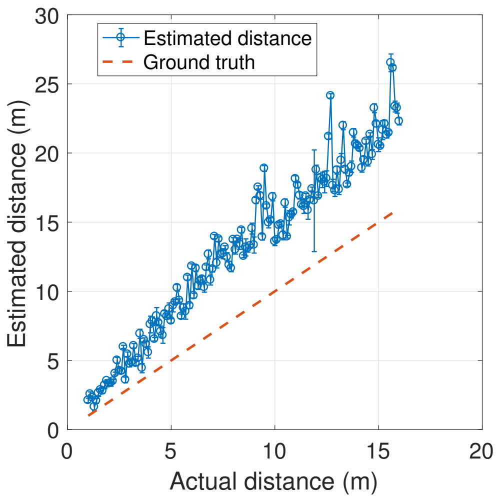
Even in this more challenging indoor environment (Fig. 3.12(a)), ranging with AP B with the higher 80 MHz bandwidth at 5GHz show meter-level accuracy, while confirming



(a) Room setting.



(b) Vary distance.

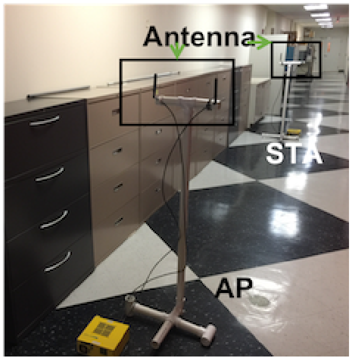


(c) Vary distance. Offset corrected

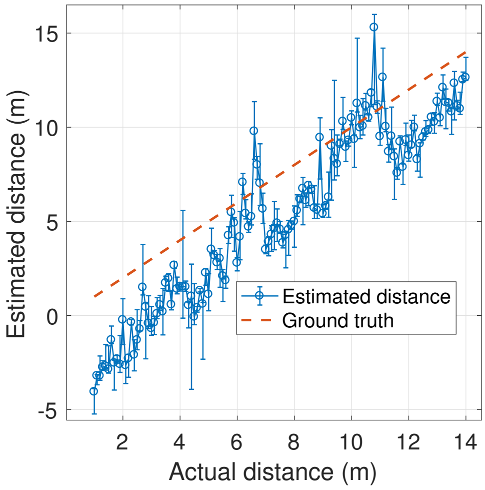
Figure 3.10: Indoor scenario: room setting (0.7 m height). This setup uses WiFi card A for both STA and AP sides.

no underestimation of the distance compared to ranging with WiFi card A.

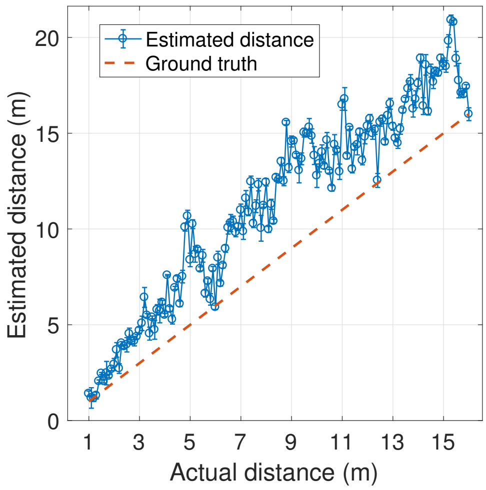
Indoor AP, outdoor STA. Indoor access points are frequently used by stations (e.g., smartphones) outdoors for positioning. To evaluate such a scenario, we start with fixing the AP location in an office inside single floor building (the office has window facing the road) while moving the STA outdoor. We test two setups: 1. moving the STA parallel to the road, 2. moving the STA across the road. Fig. 3.13(b) shows the estimated distance after correcting the offset using 6 feet cables. In such a common setup, the ranging system



(a) Corridor.



(b) Corridor (0.7 m height).

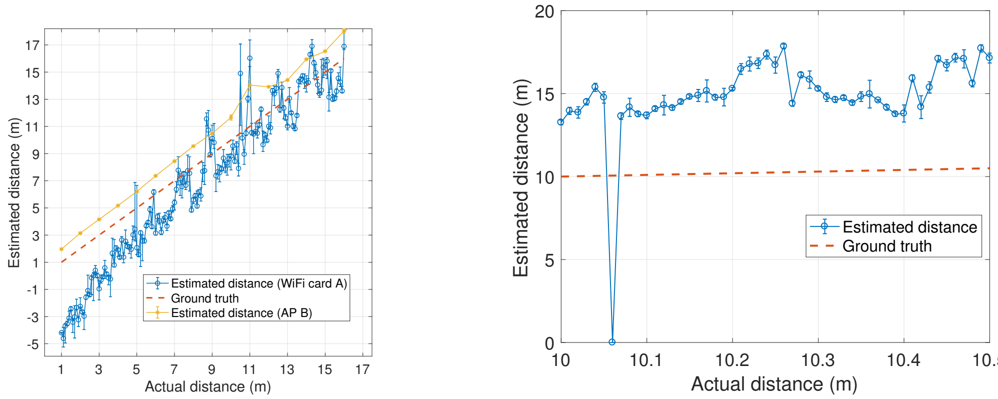


(c) Corridor (1.4 m height, offset Corrected)

Figure 3.11: Indoor scenario: corridor (1.4 m height). This setup uses WiFi card A for both STA and AP sides.

can estimate the distance while still being affected by multipath issues showing up to 3 m variations in the estimated distance for 1 m step. This is shown in Fig. 3.13(c), as the STA moves between the cars, the signal is affected heavily by multipath, and eventually encounter a total signal blockage while the STA is 28 m, and 36 m from the AP.

For indoor AP B and an outdoor WiFi card A station (Fig. 3.14), the signal is weaker than with WiFi card A as AP, likely due to the higher bandwidth and carrier frequency and hence distances above 20m, the distance estimates become unreliable. According to these results, it seems that the underestimation is an undesirable result of the algorithms in the



(a) (b)

Figure 3.12: (a) Indoor scenario (room (0.7 m height)) for comparing ranging results for using WiFi card A as AP compared to AP B while using WiFi card A as STA in both scenarios. (b) 1 cm step in indoor room (1.4 m height, offset corrected). Zero distance represents no signal.

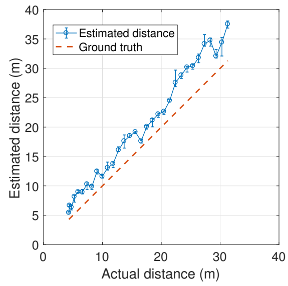
proprietary firmware.

In the next setup, we change the environment of the AP from the single-floor building to a 7-floor building while keeping the STA in nearby locations. In specific, we fix the AP in the third floor of the building near the window, with locations of the STA varying from the ground floor inside the building (right below the AP) to the outdoor field around the building. Among these setups, we also vary the orientations of antennas between vertical (antennas of each node pointing up) and horizontal (antennas pointing horizontally to the same direction). Fig. 3.13(d) shows how the orientations of the antennas can affect the estimated distance. For example, the deployment of AP and STA in different floors requires the antennas to be horizontally oriented, so that signals can propagate vertically between the floors. The bars in blue and yellow showed in Fig. 3.13(d) indicates the effect of antenna orientation on distance estimation.

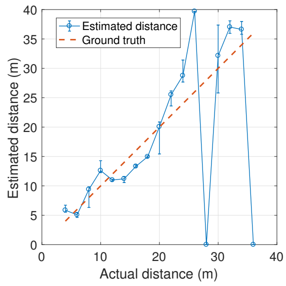
3.5 Correcting Ranging Errors

Based on our findings, we discuss in this section how to correct the ranging error using standard localization error correction techniques.

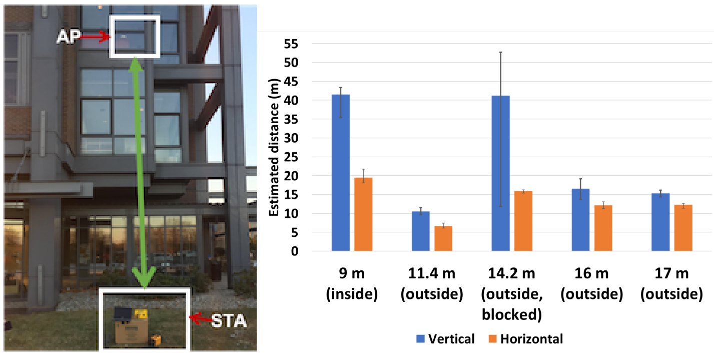
(a) 1-floor building.



(b) Parallel to road (offset corrected).

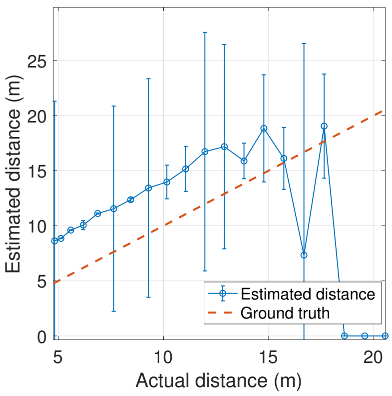


(c) Crossing the road.

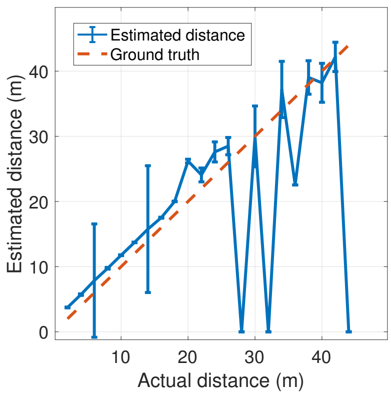


(d) 7-floor building

Figure 3.13: Indoor AP outdoor STA. Both AP and STA use WiFi card A. Zero distance represents no signal



(a) Parallel to road (offset corrected.



(b) Crossing the road.

Figure 3.14: Indoor AP (AP B), outdoor STA (WiFi card A). Zero distance represents no signal.

3.5.1 Temporal Filtering

In standard ranging systems, simply averaging multiple measurements of the same location could help to filter out hardware/software noises, but cannot eliminate the multipath effect

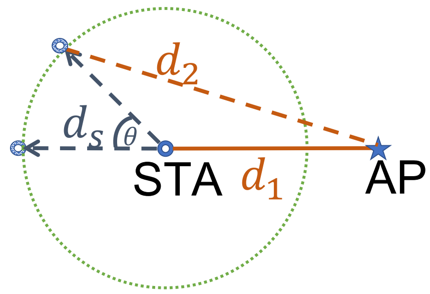


Figure 3.15: Displacement limit illustration.

which leads to distance overestimation. For example, previous work [73] has shown that using 50-percentile as an estimator leads to overestimating the distance. Thus, percentile bellow 50% is suggested. However, for this FTM ranging system deployed in environments showed in Fig. 3.7 and Fig. 3.8, we observe both underestimation and overestimation of the distances for the same location. Moreover, the overestimated distances presents significantly higher errors compared to underestimated counterpart. (Fig. 3.9).

In dynamic environments, such as highways as we shown in Fig. 3.9, moving objects could temporary block the direct path of the transmitted signals or add more reflectors that results in higher the ranging errors. Here we illustrate how standard temporal filtering techniques could improve the results for this setup.

We take a window of 10 bursts which each being 30 packets long, and analyze the data over this window. We estimate the most probable distance by building histogram for each window. It is worth mentioning that the most probable estimation is affected by the duration of the object/vehicle's affecting the ranging the system. Another approach [6] is to use clustering, assuming that the direct path results in closest estimations of the ground truth compared to the overestimated/underestimated data resulted from multipath. We sort the clusters centers and estimate the average of lowest half of the centers, while rejecting the lowest center. The clustering technique has median error of 0.2 m and 0.6 90-percentile compared to median error of 0.7 m and 2.2 m for the averaging technique. The most probable estimation has 0.3 m median error and 0.78 90-percentile while the minimum estimator has 0.4 m median error and 1 m 90-percentile.

3.5.2 Spatial Filtering

Given a fixed AP, we can leverage the mobility of a STA to collect multiple measurements over different locations. The estimated distances for these multiple locations may vary significantly as we have shown before because of multipath. These outliers can be filtered by validating the estimated distances using basic laws of geometry.

Known STA Displacement. As a STA moves from one location to another, the ranging algorithm estimates the distances from these two points to the AP. Because of multipath, synchronization, and bandwidth issues, these distances could be overestimated or underestimated. To filter out inconsistent estimations with the displacements, extra constraints need to be applied to them. The displacement of the STA can be estimated indoor using smart phones' inertial sensors [74] and outdoor using on board vehicle sensors [75].

Following this idea, as the station moves with a distance d_s, then there is a limit on the new estimated distance compared to the previously estimated distance. We formulate this limit in the following inequality:

$$|d_1 - d_s| \leq d_2 \leq d_1 + d_s \tag{1}$$

Fig. 3.15 illustrates this displacement inequality, in which, whenever the STA moves with an angle $\theta > 0$, then $d_2 < d_1 + d_s$. This limit can be proved using the triangular inequality [6]. Equality holds in this inequality when the STA moves on the line towards the AP or backwards away from the AP. Using this method, we filter and correct the estimated distances that violates the displacement inequality: If $d_2 > d_1 + d_s$, we assign $d_2 = d_1 + d_s$; if $d_2 < |d_1 - d_s|$, we assign $d_2 = d_1 - d_s$. These assignments are based on the assumption that the direction of STA is not known.

3.5.3 Evaluation

Evaluating the correction technique with different movement patterns is important. Therefore, we focus on the evaluation dataset on having common setups with different movement patterns, not only moving in the same direction along a straight line. We start with indoor scenario, in which we conduct an experiment in a cubicles office (10 m x 20 m with height

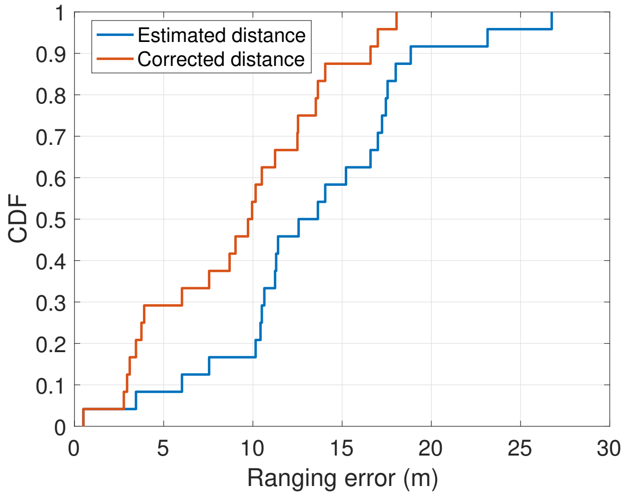
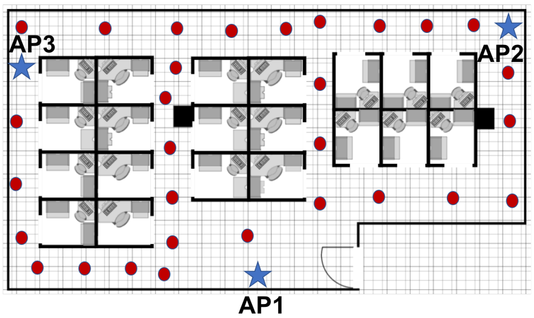


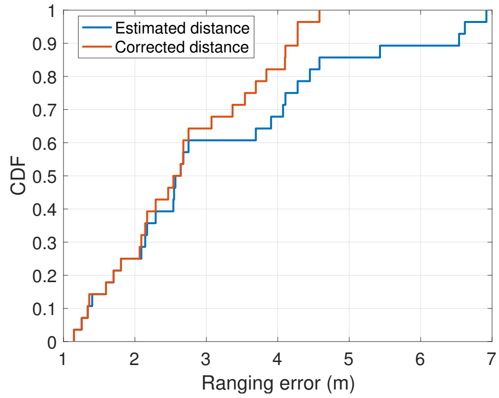
Figure 3.16: Effect of displacement limit on ranging error for indoor AP and outdoor STA

2.7 m). In this indoor setup, we fix the AP in the middle of the area and move the STA trying to cover the whole area. Fig. 3.17(b) compares the CDF of the error of the estimated distance to the corrected distance. The corrected distance achieves 2.5 m median error, and 4.78 m 90-percentile, compared to 2.6 m median error, and 6.5 m 90-percentile for the estimated distance without correction. For the second setup, we fix the AP indoors, in the third floor of a seven floors building (near the window facing the parking lot), and we put the STA on the roof of a moving car. The car traverses the whole parking lot while logging the ranging readings along with GPS readings as ground truth location. Fig. 3.16 presents the CDF for the error of the estimated distance and the corrected distance using the displacement inequality. The corrected distance achieves 9.8 m median error, and 16.5 m 90-percentile compared to 13.1 m median error, and 18.8 m 90-percentile for the estimated distance without correction.

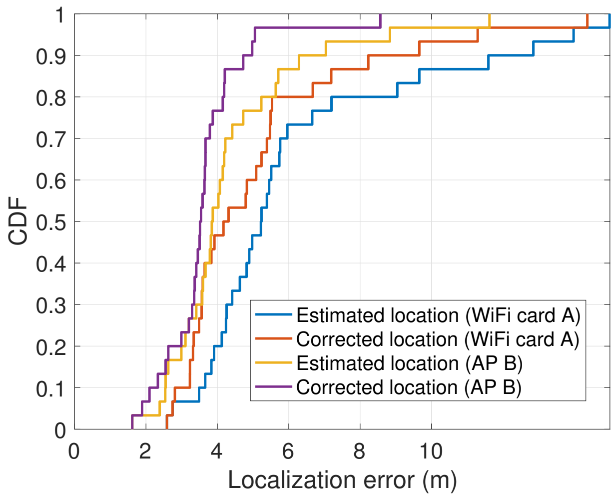
Localization error. Beside the ranging performance, we evaluate the localization accuracy indoors. Using at least three APs, a STA is able to estimate its location by trilateration using the locations of these APs, and the estimated ranges to them. We use the standard iterative nonlinear least squares trilateration algorithm [76, 77]. We fix three APs in the cubicles office (the same area used for evaluating the ranging accuracy), trying to cover the whole area with APs. In this cubicles area, there are two load bearing columns (0.8 m x 0.7 m wide), which are able to block the signal. We compare the localization



(a) Map of cubicles office.



(b) Ranging error in cubicles office.



(c) Localization error in cubicles office.

Figure 3.17: Illustration for the displacement limit and its effect of on the ranging and localization error along with a map for the indoor testbed. Red dots represents STA locations.

error between using WiFi card A or AP B as APs, while using the WiFi card A station. Fig. 3.17(c) shows that the ranging system, using WiFi card A APs, is able to localize a STA with 5.2 m median error and 11.6 m 90-percentile, while achieving 3.8 m median error and 6.2 m 90-percentile for using APs B. On the other hand, after correcting the ranging estimations for each AP, and using these ranging estimations for locating the STA, the localization error improves to 4.2 m median error and 8.2 m 90-percentile for WiFi card A setup, and to 3.5 m median error and 4.7 m 90-percentile for AP B setup.

3.6 Related Work

Evaluating multipath. Work in [78] evaluate indoor LoS scenarios, verifying the directional and polarization characteristics estimated by the the RiMax algorithm [79], subtracting the Specular Multipath Component from the observed power spectrum. In [80], they quantitatively analyze the effect of angle of inclination between the STA and AP in tracking using RADAR. Markov modelling of spatial variations seen in multipath is done in [81], and is verified by taking measurements at 60GHz in a reverberation chamber. Another line of work [82] take a geometry based approach to simulate the multipath using a nonlinear multipath filter.

Evaluation of time-based ranging systems. RADAR systems were evaluated by Derham et al. in [83], calculating the FFT of various received signals to determine the characteristics of coherent RADAR ranging signals in real conditions. GPS has been evaluated by the Naval Air Development Center in [84], where they create a setup to test every possible noise-contributing factor independent of the other.

Localization correction techniques. Tonetrack [6] implements a frequency combining algorithm (to increase the bandwidth) on the WARP hardware radio platform to track WiFi-based devices indoors. In this system, they propose a triangular inequality and clustering-based outlier detection to filter the NLoS APs. Chronos [85] proposes an indoor tracking algorithm that stitches the transmitted information over multiple bands, while leveraging a single MIMO AP. Work in [73] presents a firmware-customized time-based indoors ranging system running with a filter based on statistical learning to filter out multipath measurements.

This related work either leverage specialized hardware, or customized firmware to support nanosecond ranging time measurements. In this work, we evaluate the FTM protocol that is already standardized in IEEE 802.11-2016 [65] and being commercialized in recent WiFi chipsets [66].

3.7 Conclusion

This work presents a measurement study that evaluates a WiFi time-of-flight ranging system and verifies performance expectations. Moreover, it introduces the use and calibration of an open platform for WiFi time-of-flight ranging that future research can build on. We learned from our measurements that this ranging system is indeed capable of accurate meter-level ranging in open-space outdoor environments once calibrated. In indoor lab and office environments with multipath, both ranging and positioning (trilateration) errors increase to about 5m unless the deployment is dense enough to operate at higher bandwidths (80MHz in our experiments). This occurs even in settings where LoS reception is not blocked. We also, unexpectedly, found cases where the ranging system significantly underestimates the distance. Overall, at low bandwidth, accuracy in rich multipath environments does not seem higher than demonstrated by other positioning systems but the technology promises to deliver this accuracy with relatively few access points and less site survey overhead. With a dense deployment of access points so that multiple access points can be reached by high bandwidth signals, accuracy improves.

Chapter 4

Wi-GO: Accurate and Scalable Vehicle Positioning using WiFi Fine Timing Measurement

4.1 Introduction

The rapid evolution of advanced driver assistance and vehicle automation systems, along with their growing market [86, 87], have led to increased demand for lane-level vehicle positioning that is accurate even in urban canyon environments. Example applications for such solutions include lane-level navigation and vehicle safety communications.

Today's vehicles primarily use the Global Positioning System (GPS), often in conjunction with vehicle odometry for correcting short term GPS biases. However, in many challenging environments, such as urban canyons, bridges and tunnels, multi-path fading or shadowing considerably degrades satellite positioning accuracy. While research has shown how position accuracy can be further improved to a few meters with motion sensors and map matching [75, 88–90] in some urban environments, these still face challenges in more extreme urban canyons. To achieve lane-level positioning, highly instrumented automated vehicle prototypes use cameras or LiDAR sensors to reference their measurements against available detailed models and imagery of the roadway. Creating, maintaining, and making available such detailed image models for all roadways is laborious and resource demanding, since it can undergo frequent changes due to reasons such as snow or falling leaves.

To reduce reliance on such resource intensive image registration, we investigate whether WiFi time-of-flight ranging as specified in the WiFi Fine Time Measurements (FTM) standard [65] are sufficiently uncorrelated with GPS measurements to achieve lane-level positioning in urban canyons. While WiFi [91,92] positioning or is frequently used in smartphones, these have been based on received signal strength (RSS) positioning which is limited to an accuracy of tens of meters in urban canyons. In contrast, FTM ranging can achieve meter-level accuracy in open-space environments [66] [26]. Given this promise of improved accuracy and its wide availability, our study focuses on WiFi FTM and explores its utility in a vehicular context.

To address the lane-level urban canyon positioning challenge, this work introduces Wi-Go[1], a scalable and accurate vehicle tracking technology that complements GPS and odometry with time-of-flight measurements and multilateration to surrounding access points through the recently standardized WiFi Fine Time Measurement (FTMs) protocol. It can opportunistically use WiFi access points deployed in buildings or in cars parked along the roadway. Since the access point location is usually not known a priori, we design FTM-SLAM, a collaborative simultaneous localization and mapping approach to simultaneously track landmark (APs and parked vehicles) positions as well as moving vehicle positions. Realizing this requires addressing several key issues.

First, moving from a passive-client approach as in conventional RSSI-based WiFi positioning, to the actively-signaling-client approach used in Time-of-Flight based tracking, can introduce contention and channel congestion, particularly in densely populated areas such as urban canyons. This creates challenges when scaling to larger numbers of vehicles and access points. Consider a scenario in which hundreds of nearby vehicles send periodic FTM requests on the same WiFi channels. Since every FTM request triggers a sequence of packet transmissions specific to one client, this can easily exhaust the available channel capacity, introduce latency and therefore degrade the positioning accuracy for these clients. Note that this was not a concern for RSSI-based positioning since many clients can passively overhear the same access point message but due to the lack of synchronized clocks FTM requires round trip packet exchanges. To tackle this issue, Wi-Go mitigates channel congestion by optimizing the FTM message rates to each AP to maximize the tracking accuracy, while remaining below a cumulative allowed message rate in a given region.

Second, the FTM ranging process involving a series of packet exchanges introduces time offsets between the range measurements to individual access points, while standard multilateration method in wireless localization assume quasi-simultaneous ranging to multiple

[1]Wi-Go stands for WiFi, GPS and odometry based tracking

landmarks. This is particularly important in the vehicular context where a fast moving vehicle can travel a significant distance between individual measurements. To address this challenge, we propose *Mobile Multilateration*, a novel tracking algorithm, to track a moving vehicle with individual ranges obtained at different positions, while still leveraging odometery information to relate each position of the vehicle to its previous position.

Third, to minimize receiver complexity, it requires simultaneous localization and mapping (SLAM) techniques that only use range information while existing SLAM work heavily relies on bearing measurements as well (e.g., [93, 94]). Wi-Go presents a range-only SLAM framework suitable for multilateration with Fine Time Measurements; it explicitly represents the uncertainty of both the position estimates via particle filters, and can use FTM measurements from a second antenna to resolve AP position ambiguity on linear roadways.

Fourth, in environments such as urban canyons, WiFi communication can also be heavily affected by multipath fading and shadowing. Simply using WiFi measurements that are potentially low quality will not yield the desired accuracy improvements over GPS. Therefore we design an uncertainty weighted multilateration technique that estimates and explicitly considers the quality of the current GPS and FTM measurements when updating the vehicle and access point location.

We prototyped Wi-Go with Intel 8260 wireless cards on a small form factor PC installed with roof-mounted antennas in a vehicle and implemented backend algorithms in a cloud server. We evaluated Wi-Go in two deep urban canyon environments of upper Manhattan as well as in a suburban residential setting. Wi-Go achieves 1.3 m median error for vehicle tracking, with relatively low AP density (4 APs), in our urban canyon dataset in which a baseline of a built-in GPS only achieves 9.04 m median localization error. Meanwhile, Wi-Go's adaptive algorithm can efficiently adapt the FTM message rate while still showing 41.7% improvement in terms of vehicle localization error.

Summary of Contributions. Wi-Go and its FTMSLAM algorithm make the following contributions compared to earlier positioning and SLAM algorithms:

- Designing a novel FTM-SLAM algorithm to intelligently complement GPS and odomentry with range-only WiFi FTM data, while addressing the resulting latency and multi-path challenges.

- Mitigating channel congestion due to active time-of-flight ranging messages, by intelligently distributing the cumulative allowed message rate over the nearby APs.

- Introducing the opportunistic use of parked vehicles for positioning by incorporating pairwise-distance estimates between APs in parked vehicles.

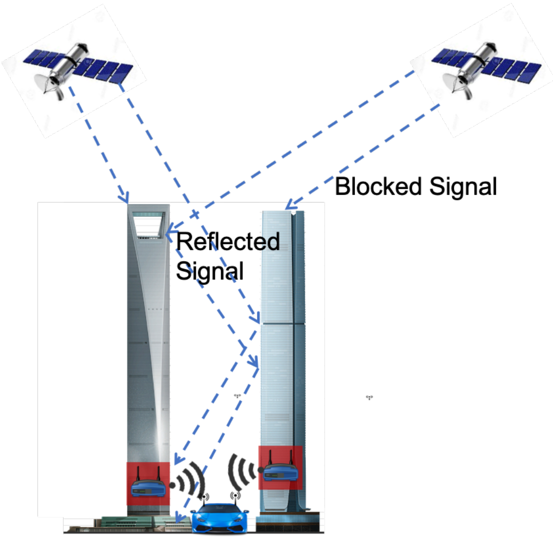
- Demonstrating and validating that the Wi-Go system with its use of WiFi FTMs meets lane-level positioning requirements through extensive experiments in two challenging urban canyon environments and a suburban setting, with a small fleet of 5 research vehicles.

4.2 Background

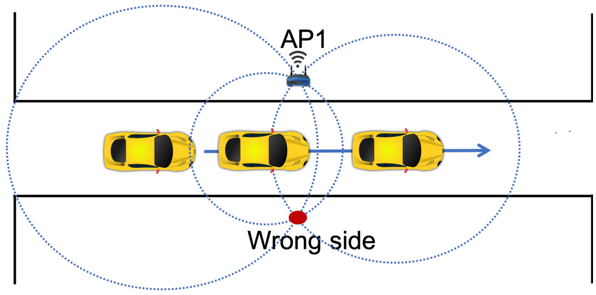
In this section, we will review the WiFi Fine Timing Measurement, SLAM, and vehicle sensing concepts that this work builds on. WiFi positioning has evolved from initial fingerprinting [8] approaches to using Angle-of-Arrival [4], dead reckoning [95] and Time-of-Flight (ToF) [96]. The ToF-based WiFi distance measurements have recently been incorporated in WiFi standards, and promise widespread AP deployment support.

WiFi Fine Timing Measurement (FTM). IEEE 802.11-2016 Standards [65] has included the Fine Timing Measurement protocol, 802.11REVmc, to perform wireless ranging by measuring the round trip time (RTT) between an AP and a WiFi station (STA). The protocol subtracts processing times from the round trip time, converts it into a one-way time-of-flight estimate, and uses this to estimate range using typical propagation speed. As an interactive protocol, it conducts multiple message exchanges to achieve a higher ranging accuracy. Recent research [26] has confirmed that the FTM protocol can achieve meter-level accuracy in open space environments but the accuracy degrades in high multipath environments.

Android [67] and major vendors of WiFi chipsets have started to support the FTM protocol. However, their accessibility to PHY layer information like Channel State Information is quite limited [97] since the station has to associate with the AP, which takes



(a) WiFi Vs GPS



(b) Vehicle linear movement

Figure 4.1: (a) WiFi edge over GPS. (b) Effect of linear movement of a vehicle on AP localization error.

several seconds [98]. This is not suitable for fast moving vehicles.

SLAM. Classic SLAM frameworks like FastSLAM [93, 99, 100] require ranging technology (e.g. LiDAR, or stereo cameras) that estimates the distance and bearing to a landmark. Range-only SLAM with RF or sonar beacons [101–105], approximates the vector connecting current pose with current location estimate of landmark, requiring the initial landmark location estimate to be accurate. Previous range-only SLAM frameworks used in robotics [102, 103, 106] has initialized landmarks through taking majority votes over multiple solutions. Each of the solutions either fits a pair of ranges (assuming the robot does not move in a straight line) or draws probabilistic samples from a circle around the robot location with radius equal to the range. These two initialization approaches may converge to the actual location and achieve high accuracy if the robot is not moving in a straight line, which is not the case in our application (as shown in Fig. 4.1(b)).

On the other hand, current WiFi SLAM [94, 107–109] either treats RSSI fingerprints as landmarks and tracks them simultaneously with users location, or augments WiFi with cameras to estimate bearing to the RSSI-based fingerprints. None of these prior works have estimated the APs' locations simultaneously while tracking a user, because current propagation models that map RSSI to distance are not accurate enough and cannot generalize to different environments. WiGo introduces a new SLAM approach called FTMSLAM, to track APs and vehicles with range-only FTM range measurements. FTMSLAM uses novel opportunistic sensing algorithms including vehicle to AP bearing estimation, adaptive FTM range calibration learning, and vehicle location correction through APs street maps.

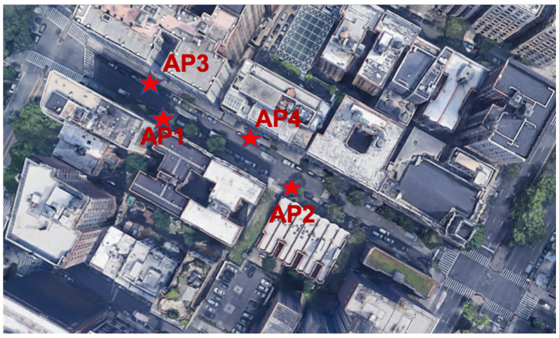
On-board vehicle sensors. Modern vehicles are equipped with sensors to measure vehicle speed, steering wheel angle, heading, yaw rate, etc. While this information is usually communicated on the CAN bus (and often readable from the On-Board Diagnostics (OBD-II) port), its encoding is specific to certain vehicles and proprietary [110]. In this work, we leverage available on-board sensors to feed our FTMSLAM framework with precise odometry. Moreover, we leverage this odometry information as an independent, orthogonal source for correcting FTMs in over- and under-estimate cases.

4.3 Can WiFi Augment GPS?

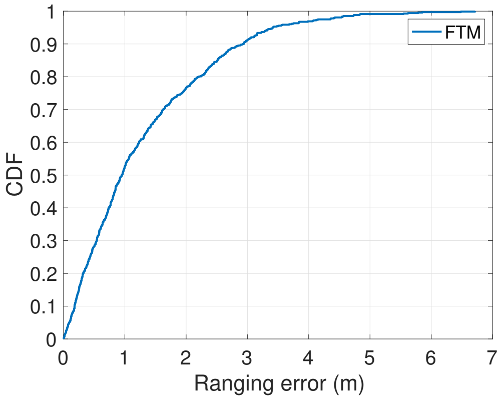
Our Wi-Go system design is motivated by the observations that urban canyons, a key environment with degraded GPS, usually enjoy a dense deployment of WiFi Access Points (APs) (as shown in Fig. 4.1(a)). Current WiFi outdoor localization [12, 42] mainly counts on RSSI fingerprinting, which is shown to be less accurate due to several innate limitations, like multipath effects [8], variation of performance between different devices of the same vendor [111], and low granularity [112].The recently available WiFi Fine Time Measurements, instead, promise a more robust, accurate, and fine-grained ranging measurement to enable mobile multilateration techniques, suggesting a possibility that WiFi FTM ranging could further complement GPS.

To examine our hypothesis, we conduct a preliminary experiment in an urban canyon environment (Manhattan, NYC). In this experiment, we place 4 access points on top of 4 parked vehicles as 'virtual' WiFi APs, and evaluate the ranging and location error, in the rest of the work, using the following four different technology options:

(1) **WiFi FTM baseline**. Our WiFi AP is equipped with an Intel Dual Band Wireless-AC 8260 chipset that supports FTM capability; (2) **Standalone GPS**. Our vehicle is equipped with an after-market U-blox EVK-7P GPS receiver (<1 m precision in an open



(a) Urban Experiment



(b) CDF

Figure 4.2: WiFi FTM ranging accuracy.

Approach	Vehicle GPS	GPS	Android Loc.
Error (m)	9.04	18.2	19.6

Table 4.1: Median tracking error of related technologies in Manhattan, NY.

sky setup); (3) **Vehicle GPS**. Meanwhile, our vehicle retrieves vehicle GPS readings from OBD-II port. This built-in GPS has been internally corrected using the vehicle IMU and odometry sensors; (4) **Android Fusion Location API**. Finally, we collect location measurements through Android Fused Location Provider API from a Google Pixel 3a smartphone that is placed on the vehicle's dashboard. The Android Fused Location Provider API fuses multiple sensors including GPS, WiFi RSSI, and cell tower positioning together to provide localization information. These technologies summarize the state-of-the-art outdoor localization approaches which are either GPS, motion sensors dead-reckoning, or based on WiFi RSSI, Cellular RSSI, or a fusion of some of them.

In order to acquire ground truth locations of the vehicle, two Intel RealSense Depth Cameras are mounted on both sides of the vehicle's roof that log depth information of both sides of the road. We examine the recorded frames looking for landmarks such as light poles and trees and use these landmarks' positions to infer the vehicle's position. Specifically, when the landmark appears in the middle of the field of view, we use the landmark's location along with its depth to compute the current vehicle location.

Fig. 4.2(b) shows the ranging error of only WiFi FTM as other competing technology

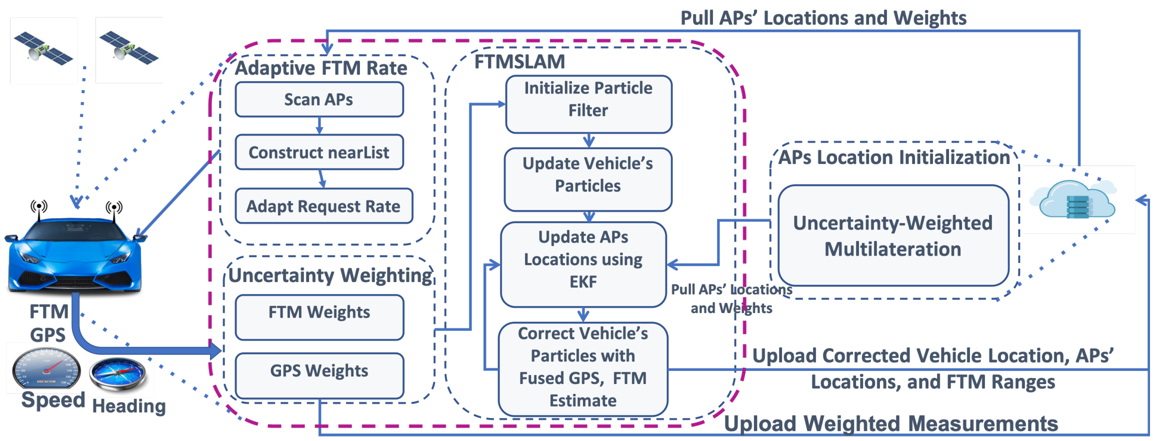


Figure 4.3: Wi-Go System Architecture Overview.

options do not report range measurements. The ranging error of WiFi FTM is 0.95 m median error (and 2.9 m at 90-percentile CDF curve). On the other hand, we also analyzed the localization errors of vehicle GPS, Android Fused Location, and standalone GPS as shown in Table 4.1. This preliminary result indicates that WiFi FTM seems to hold great promises of complementing and augmenting existing outdoor localization techniques.

In the next sections, we describe in detail the design of Wi-Go, which achieves this promise of lane-level positioning.

4.4 The Wi-Go System

The objective of Wi-Go is to provide a practical positioning system that scales in real world scenarios where tens of APs and hundreds of vehicles compete in the shared WiFi channels through active WiFi FTM ranging. Besides scalability, we aim for uninterrupted meter-level positioning by augmenting WiFi FTMs with GPS and odometry readings.

4.4.1 System Overview

Incorporating WiFi FTMs for outdoor positioning is challenging due to the following reasons: First, with extended FTM ranging latency, a fast vehicle could register multiple locations at different nearby APs, imposing a vehicle tracking challenge; second, FTMs are

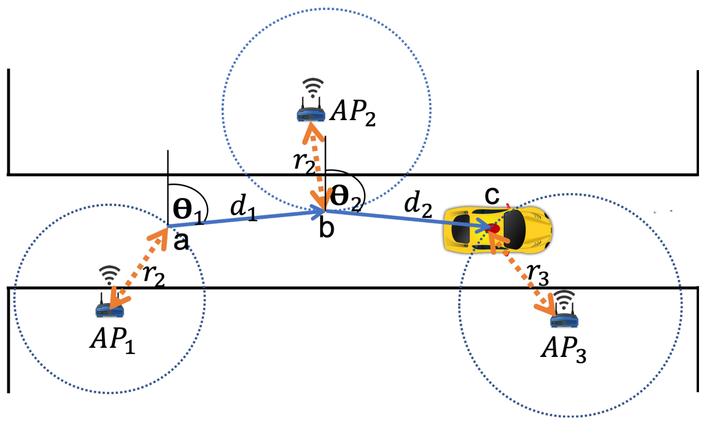


Figure 4.4: Illustration of our Mobile Multilateration. A vehicle moves from point a, to b, then from b to c.

affected by multipath and are a range-only measurement that does not offer bearing information, which makes jointly locating APs and vehicles challenging; third, the injection of FTM packets from many vehicles onto the same WiFi channels causes network congestion;

To address the above challenges, we design Wi-Go as shown in Fig. 4.3. In each participating vehicle, Wi-Go collects WiFi FTMs from the surrounding APs, along with the timestamp, GPS reading, and the vehicle's speed and heading through on-board sensors. We assume that vehicles will at least occasionally enter open-sky GPS conditions, and that this could be used to establish position in the world coordinate frame. The Wi-Go system then tracks a vehicle by starting with an initial location estimate, gradually refining based on successive GPS readings, vehicle odometry, and WiFi FTM range measurements to access points positions. It uses simultaneous localization and mapping techniques to jointly estimate vehicle position and access point positions. As in current WiFi positioning systems, access point position estimates can be shared across vehicles through a server.

WiGo addresses the fast moving vehicle challenge through mobile multilateration, compensates for multipath through uncertainty-aware fusion, and controls FTM request through a congestion-aware optimization. We describe these techniques in the following subsections.

4.4.2 Mobile Multilateration

Tracking a rapid-moving vehicle is not an easy task, since it requires quick ranging due to vehicle speed. If a vehicle moves at 20 m/s and the standard FTM latency is 0.2 s, then the next ranging measurement will be at least 4 meters away. A new form of multilateration is thus needed for a moving vehicle to collect sequential ranging measurements at different precise locations from multiple nearby APs.

Fig. 4.4 illustrates this issue: A vehicle starts ranging at location A and obtains a ranging measurement r_1 from AP_1, then it gets r_2 from AP_2 at location B, and finally it collects r_3 from AP_3 at location C. Our goal is to find the current location of the vehicle $\mathbf{v}$, given APs locations [$\mathbf{x}_1$, $\mathbf{x}_2$, ..., $\mathbf{x}_n$] and corresponding ranges [r_1, r_2, ...,r_n]. To tackle this issue and relate the measurements of location A and B to location C, we leverage vehicle odometry readings to estimate displacement and heading between these different locations. Formally, we formulate this problem of Mobile Multilateration as a non-linear least squares formulation:

$$\begin{aligned} &\underset{v_n}{\textbf{argmin}} && \sum_{i}^{n} (||\mathbf{x}_i - \mathbf{v}_i|| - r_i)^2 \\ &\textbf{subject to} && v_i = reckon(v_{i-1}, d_{i-1}, \theta_{i-1} + 180), \\ & && (\text{where} \quad i = 1, ..., n) \end{aligned} \tag{1}$$

Through this formulation, we could identify the current location of the vehicle by minimizing the least square error while backwardly holding the constraint functions. For instance, with the last displacement and heading, we can derive the current location v_n from the preceding location v_{n-1}.

4.4.3 Uncertainty Weighting

During the collection of measurements, weights are assigned to FTMs and GPS readings, representing the confidence of these readings, which will be used later in the fusion as described in Section 6.

FTM Weighting. We use vehicle wheel encoders and inertial sensors to assign weights to FTM measurements. In this step, we do not correct collected range measurements; instead, we simply assign weights to these measurements and feed them along with their weights to the FTMSLAM algorithm (Section 6). Triangle inequalities are used to evaluate measurement confidence: in Fig. 4.5(a), let d be the estimated displacement distance

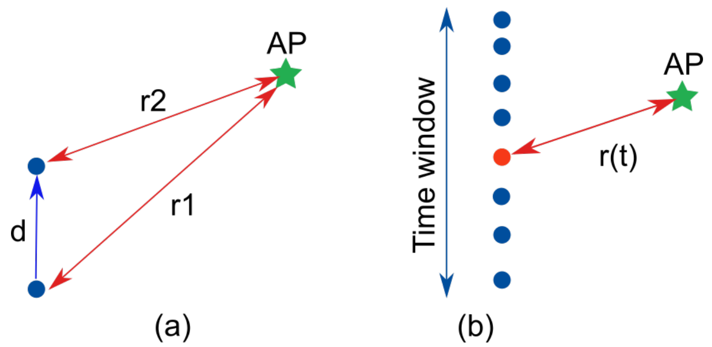


Figure 4.5: FTM Weighting. (a) Triangular inequality. (b) Assigning measurement confidence based on a time window.

reported by the inertial sensors between two FTM measurement locations. Also, let r_1 and r_2 be the measured FTM ranges at these two locations. Then the following triangular inequality must be satisfied: $|r_1 - r_2| \leq d \leq r_1 + r_2$. For each FTM measurement r(t), we evaluate the triangle inequality of r(t) with another FTM measurement within a small time window (Fig. 4.5(b)). The weight $w^{FTM}(t)$ for measurement $r(t)$ is thus the ratio of measurement pairs that satisfy the triangle inequality over the total number of pairs.

GPS Weighting. GPS NMEA (National Marine Electronics Association) sentences are used to obtain data which is leveraged to assign a weight for the GPS location. For a given time t, this data includes: signal-to-noise ratios $SNR(t)$ for each observed satellite, number of observed satellites $n(t)$, GPS quality $q^{GPS}(t)$, and horizontal dilution of precision $HDoP(t)$. GPS quality indicates 3 different states: no fix ($q^{GPS}(t) = 0$), GPS fix ($q^{GPS}(t) = 1$), or differential GPS fix ($q^{GPS}(t) = 2$). $HDoP$ measures the geometric quality of GPS satellites configuration in the sky [113]. The smaller the HDoP number, the better the geometry in terms of being spread in the sky, which has been reflected into the GPS location precision. The GPS weight for the system is calculated as $w^{GPS}(t) = \frac{q^{GPS}(t)*n(t)*mean(SNR(t))}{HDoP(t)}$.

4.4.4 Congestion-Aware Adaptation of FTM Request Transmission

When many vehicles try estimating their positions using requests and responses from multiple APs nearby, WiFi channel congestion may occur. Since FTM is an active measurement, it is initiated from the STA and consumes bandwidth for every burst it sends out. With hundreds of vehicles and tens of APs sharing limited WiFi channels, it is critical to limit how many FTM requests a vehicle can send out under fixed bandwidth restriction. This challenge is addressed in this section by adapting the samples per burst parameter (spb).

Problem Formulation. Thanks to pioneering studies on congestion control in vehicular networks [114–116], we assume that the vehicle estimates the maximum message rate limit that it is allowed to send without causing congestion. In this paper, the main task is thus to distribute and provide message rates for each vehicle to different APs nearby, while still honoring the cumulative message rate limit and minimizing the vehicle location error. Note that the messaging between vehicles and APs consumes channel bandwidth and that potentially affects APs as well. To address this, specific channels or partial bandwidth could be reserved for FTM messaging or a bandwidth limit can be set for such messaging. This will in turn determine the maximum message rate per vehicle.

To achieve this goal, as shown in the objective function (Eq. 3), we aim to minimize the key factors that contribute to vehicle localization error collectively: the geometry of the chosen subset of APs $(1 - \delta(\mathbf{spb}))$, error model in AP location (e_{AP}), and FTM ranging error e_r (reflected in the error covariance matrix $\mathbf{C}_{\Delta r}$).

Approach. A naive solution would be to distribute rate limit equally over surrounding APs; however, not all APs in range can precisely estimate a vehicle's location. Therefore, it is important to determine the subset of nearby APs the FTM requests should be sent to. Collinearly distributed APs, for instance, fail to provide an accurate position estimation; similarly, APs that are in distant locations provide range measurements with larger errors.

Motivated by the above observations, we instead use a weighted round robin schedule for ranging to each nearby AP; in particular, we intentionally adapt the samples per burst parameter for each AP on one hand, while maximizing the vehicle localization accuracy on the other hand. For an AP, the spb can be set to zero if this AP is unlikely to be useful, and the sample rate adapts based on the location of a vehicle and its AP counterpart, respectively.

Given that $\mathbf{x}$ is the AP's location estimates, $\mathbf{v}(t-1)$ is the last location estimate for the vehicle, and $\mathbf{r}(t)$ is the vector of range measurements to these APs, the ranging model would be $\mathbf{r}(t) = distance(\mathbf{x}, \mathbf{v}(t)) + \mathbf{e}$. $\mathbf{e}$ represents the total error originated from FTM ranging errors $\mathbf{e}_r$ plus errors in AP's locations ($\mathbf{e}_{AP}$). As distance is not a linear function of locations of APs and a vehicle, we can linearize these equations using initial estimates of $\mathbf{x}$ and $\mathbf{v}(t)$. Through this linearization, we can determine corrections to the estimates and obtain current location of the vehicle: $\Delta\mathbf{r} = \mathbf{A}\Delta\mathbf{v} + \mathbf{e}$, where $\mathbf{A}$ is an nx3 matrix of the partial derivatives (Jacobian) of the distance function with respect to the unknown vehicle locations. Through a standard least squares solution ($\Delta\mathbf{v}$), the covariance matrix of the solution is:

$$\mathbf{C}_{\Delta\mathbf{v}} = (\mathbf{A}^T\mathbf{C}_{\Delta\mathbf{r}}^{-1}\mathbf{A})^{-1} \tag{2}$$

where $\mathbf{C}_{\Delta\mathbf{r}}$ represents the error contributed by each AP. As this error is dependant on the location of AP (e.g., some APs could be more affected by multipath than others), these errors cannot be assumed to be the same across different APs. We therefore only assume the errors of APs are independent, and the standard deviation of the total error for each AP can be calculated as $\sigma = \sqrt{\sigma_{e_{AP}}^2 + \sigma_{e_r}^2}$.

To optimize the vehicle localization error, we derive our objective function from the covariance matrix of the least squares solution (Eq. 2) as follows:

$$\begin{aligned}
\underset{\mathbf{spb}}{\mathbf{argmin}} \quad & ((1-\delta(\mathbf{spb})) \odot \mathbf{A}^T\mathbf{C}_{\Delta\mathbf{r}}^{-1}\mathbf{A})^{-1} \\
\textbf{subject to} \quad & \sum spb_i = rate_{limit} \\
& spb_i \geq 0, \qquad 1 \leq i \leq |APs|
\end{aligned} \tag{3}$$

where $\delta(\mathbf{spb})$ has the same form as the Delta function and $\mathbf{e_{AP}}$, $\mathbf{e_r}$ represents the error of APs' location estimations and range measurements respectively. In this model, prior AP location estimates are assumed to exist already and each estimate has its error derived from the covariance of that estimate; at the same time, newly observed APs are be ranged occasionally through a separate process in order to establish an initial position estimate.

The error model used for FTM range estimates can be estimated empirically by fitting the linear regression function ($e_{r_i} = a \times dist_i + b$) using some real world data [26]. The ranging error is approximated using the Central Limit Theorem when *spb* samples are taken and averaged. For large *spb*, the variance of that error can be approximated as $\frac{\sigma^2}{spb_i}$: $\sigma_{e_r} = \sqrt{\frac{(a \times distance(\mathbf{x},\mathbf{v}(t-1))+b)^2+\sigma_{emp}^2}{spb_i}}$. So the ranging error to a certain AP is estimated based on its last estimated distance and the empirically obtained standard deviation σ_{emp} of these measurements.

4.5 The FTMSLAM Algorithm

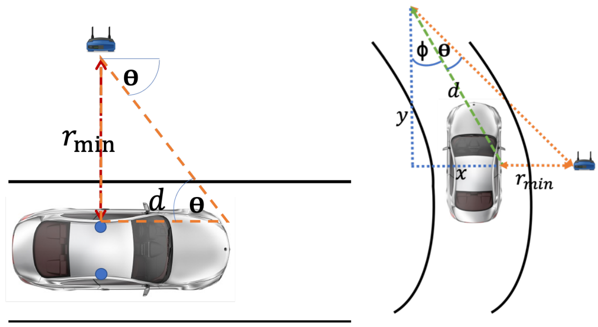
All of these ideas are put together in the FTMSLAM algorithmic framework, the cornerstone of the Wi-Go system. In FTMSLAM, the vehicle and the surrounding WiFi APs are localized and tracked, by incorporating WiFi FTMs, GPS, and on-board sensor measurements.

Novelty. FTMSLAM utilizes range-only measurements (FTM), in direct contrast to conventional SLAM approaches which require both range and bearing measurements. To conquer this challenge, we intentionally take advantage of a suite of novel methods, *vehicular opportunistic sensing*, to update and correct the locations of vehicle and APs respectively. The opportunistic sensing methods include (1) vehicle to AP bearing estimation (Section 6.2), (2) resampling particles from FTM multilateration to refine estimated location (Section 6.3), and (3) adaptive FTM range correction (Section 6.3).

4.5.1 Location Modeling and Initialization

Location Modeling. Inspired by classic FastSLAM [100], we model the vehicle location using particle filters, which provides a non-parametric probabilistic position representation. Through this model, we update a vehicle's particles through dead reckoning, and then further correct the distribution of particles using the fused location estimate of GPS and WiFi FTM measurements. At the same time, we model the AP location as a Gaussian distribution. In our study, it is found that the lack of angular information in FTM measurements introduces several new challenges: the requirement for accurate initialization, the need for fusing GPS and FTM measurements for the correction step, and the demand to estimate a range calibration factor for different APs. These issues are addressed in Sec 6.2.

AP Location Initialization. This is done through pre-collected, crowd-sourced GPS traces along with FTMs. We estimate the locations of APs using uncertainty-weighted mobile multilateration (Section 4). To be specific, the optimization problem is formulated



(a) (b)

Figure 4.6: Angle estimation illustration.

as follows: given the location $v_i(t)$ of vehicle i at timestamp t, the collected FTM range $r_{ij}(t)$, and their weights $w_{ij}^{FTM}(t)$, the goal is to identify the location of unknown AP x_j. To derive the accurate location information, we compensate the length of wired cables used to extend WiFi antennas by subtracting c from all the FTM ranges $r_{ij}(t)$. We thus solve this mobile multilateration problem through the following weighted non-linear least squares formula:

$$x_j \sum_{i}^{N_v} \sum_{t}^{T} w_{ij}^{FTM}(t)(||\mathbf{x}_j - \mathbf{v}(t)|| - (r_{ij}(t) - c))^2 \qquad (4)$$

4.5.2 Opportunistic Bearing Estimation

Using dual antennas connected to a WiFi transceiver placed on the vehicle, the bearing is estimated from the vehicle to an AP. As illustrated in Fig. 4.6, two antennas are fixed on the right and left edges of a vehicle to obtain differential measurements across the direction perpendicular to the vehicle bearing. In practice, nonetheless, the noise in FTM readings could exceed the actual difference between two measurements (left antenna and right antenna), which is highly depending on the width of the vehicle. Even worse, FTM protocols currently implemented on commercial WiFi transceivers do not grant access to any PHY-layer information that could help estimate the angle of arrival (e.g. signal phase).

To overcome this practical challenge, we consider the fact that we can estimate which side the APs are placed as well as the minimum range r_{min} to that AP over the first round of FTM measurements. Later, when we have another estimated minimum range through a new visit to this AP, we take the weighted average between the new estimate and the historical estimates to improve estimation accuracy. In the subsequent rounds, we can estimate the bearing information to that AP, whenever we reach the nearest point to the AP (at this point, the bearing should be either 90 deg or 270 deg from the vehicle's heading). As the vehicle moves forward, we can estimate the distance d from the nearest point to the AP by integrating the current vehicle's speed. With r_{min} and d, we can estimate θ as illustrated in Fig. 4.6(a): $\theta_i = tan^{-1}\frac{r_{min}}{d_i}$. This equation can be generalized for any road shape as illustrated in Fig. 4.6(b). Given r_{min}, d, and ϕ, we can estimate θ: $\theta_i = tan^{-1}\frac{x+r_{min}}{y} - \phi$, $x = sin(\phi) * d$, $y = cos(\phi) * d$.

In this algorithm, we heavily rely on the minimum range to the AP, which is attributed to the fact that FTMs are more accurate over short distances. As a result, the estimated bearing through the algorithm is relatively accurate; however, the accuracy could decrease as d increases, since the accumulation of speed sensor error negatively affects the estimated d. To estimate the distance from the vehicle to the AP more accurately, we combine the two FTM measurements from the right and left antennas respectively through a weighted average using the FTM weights.

4.5.3 FTMSLAM Tracking Framework

FTMSLAM Initialization Step. FTMSLAM requires initializing both the vehicle's location estimate and the APs' location estimates first. To do so, we use a GPS measurement with a sufficiently large GPS weight, or alternatively, through WiFi FTM multilateration with sufficiently accurate estimated locations of surrounding APs.

Update/Prediction Step. In this step, we update the locations of the vehicle and the currently discovered APs. To update a vehicle's location, we estimate its displacement from its previous location, together with the vehicle's heading using its in-vehicle sensors (these sensors provide vehicle kinematic information such as speed, steering wheel angle, heading, and yaw rate). Here, the displacement can be estimated as follows: $d = \frac{1}{2}(s_t - s_{t-1})\Delta t$. Through estimating both displacement and azimuth, which is the heading with respect to the north, we can update the vehicle's location using Vincentys formula [117] which takes the Earth curvature into account.

In the meantime we also update the locations of surrounding APs, through Extended

Kalman Filter (EKF) [100]. Given that the bearing to APs is not always available, we follow a range-only SLAM approach for opportunistically updating APs if the bearing estimation becomes unavailable. Here the location of the AP is updated, by using the vector linking the current estimate of vehicle's location with previous estimate of this AP location. Since this vector depends on the AP location initialization, an accurate initialization is crucial for such a range-only approach. To further improve the accuracy, the bearing estimate is improved when the bearing estimation algorithm is invoked.

Correction Step. In this step, we update the weight of each particle. By correcting the particle's distribution of the vehicle using both GPS and FTMs, we update the location of the vehicle and avoid the accumulation of motion sensors error. There are a variety of different places: in rural or suburban areas where the GPS is accurate but WiFi is not available, and the reverse could be true in urban canyons where GPS reception is poor, but dense APs have already been deployed. It is likely that we can correct the distribution of particles based on fused correction weight between GPS and FTM estimate.

With dense deployment of APs, we can estimate the current vehicle's location using weighted multilateration by using FTMs, their weights ($w_j^{FTM}(t)$) and current location estimates of these APs. Based on the distance between each particle and the current FTM estimate for the vehicle's location, the FTM weight of that particle is updated, $w_{part_i}^{FTM}(t)$. However, this dense enough deployment of APs is not always available. In this case, the weight of the particle is updated based on the average of the error between range measurements and estimated range, which is the distance between the particle and the current location estimate of each AP. Similarly, the GPS weight of that particle $w_{part_i}^{GPS}(t)$ is updated based on the distance between current GPS estimate and the particle.

To fuse these two weights, we take the weighted average of these two particle weights: $w_{part_i}(t) = w^{FTM}(t) * w_{part_i}^{FTM}(t) + w^{GPS}(t) * w_{part_i}^{GPS}(t)$. As the FTM weight of that particle is estimated by leveraging FTM measurements to multiple APs, we estimate this as the average of these FTM weights, so $w^{FTM}(t) = \sum_j^N w_j^{FTM}(t)/N$, where N is the number of currently seen APs.

FTM Resampling. In urban canyons, a vehicle exhibits degraded GPS readings, leading to the particles being tens of meters away from its actual location. This requires an aggressive and fast way of re-weighting the particles when the vehicle starts to observe a sufficient number of APs for multilateration. In such cases, when the current estimated vehicle location through FTM is far away from the last estimated location, all particles will end up with zero weights. To resolve this particle deprivation problem, we re-sample a small portion of the particles from the FTM multilateration location.

Adaptive FTM Range Correction. FTM weighting cannot mitigate multipath effects by itself, as multipath error in certain areas can be consistent and ubiquitous. An adaptive mechanism for correcting FTM range is thus needed to mitigate multipath effects.

In each update step, FTMSLAM uses the error between the estimated range (FTM) and the corresponding distance between an AP and a vehicle to update AP location. After multiple iterations, the accuracy of AP location estimate is seen to improve, leading to a better estimation of vehicle location. Inspired by this observation, we could calculate a range calibration factor to compensate multipath error, since the distance error to an AP would eventually converge to its actual range error. In practice, a 2D spatial map of the range calibration factor for each AP is built by clustering locations with similar patterns of multipath effects. Hence similar correction values and estimated calibration factors could be used to compensate similar multipath effects.

4.6 Parked Vehicles as Pseudo APs

In addition to opportunistically using stationary WiFi access points in buildings or roadside infrastructure that support the WiFi FTM standard, Wi-Go can be extended to also take advantage of WiFi devices in parked vehicles as Pseudo WiFi APs. As a vehicle parks, it can switch from WiFi station (STA) mode to AP mode through relatively straightforward WiFi firmware changes, if vehicle battery management system permits. Using parked vehicles as pseudo APs could effectively increase the density of WiFi APs, and also add high-quality pseudo APs since parked vehicles at the street curb are likely to have a line-of-sight propagation to moving vehicles. This innovative idea, nonetheless, will change several design choices we outlined in Section 4-6. In this section, we intend to briefly discuss these issues and also shed light on their high-level solutions.

Pseudo AP Location Initialization. The last estimated location of a parked vehicle (just before the ignition switches off) could serve as an initial location for that pseudo AP. More importantly, these parked APs can estimate distance to surrounding APs, before switching to a AP mode. This method can effectively estimate the pairwise distances between a subset of APs, which helps jointly initializing APs locations. We designed a different optimization method for parked, stationary vehicles because parked vehicles can obtain pair-wise rang measurements between them. Specifically, we can solve the multilateration problem jointly for all N_{AP} APs, instead of solving the problem independently for each AP:

$$\begin{aligned} \underset{x}{\textbf{argmin}} \qquad & \sum_{j}^{N_{AP}} \sum_{i}^{N_v} \sum_{t}^{T} w_{ij}^{FTM}(t)(||\mathbf{x}_j - \mathbf{v}(t)|| - (r_{ij}(t) - c))^2 \\ \textbf{subject to} \quad & ||\mathbf{x}_j - \mathbf{x}_k|| = r_{jk}, \qquad k = 1, 2, ..., N_{AP} \end{aligned} \tag{5}$$

Pseudo AP Location Modeling. Parked vehicles bring another challenge regarding modeling of pseudo AP location: Regular APs inside of buildings rarely change locations (maybe once a year); now, with these parked vehicles, pseudo APs could move more frequently (multiple times per day). To tackle it, we model pseudo AP location probabilistically using a particle filter approach that takes vehicle mobility into account.

Vehicle Battery Duty Cycle. Powering an AP while a vehicle is parked, may raise a battery issue if the pseudo APs are kept on for a long duration. To tackle that, we could either develop an intelligent power management algorithm that relies on Bluetooth Low Energy to wake up WiFi AP unit, or adopt a simple solution to stop vehicles from switching to Pseudo APs if its battery is less than a pre-set threshold (e.g. 80%).

The design and implementation of this idea merit an independent study, and we leave it for future work.

4.7 Evaluation

4.7.1 Experimental Setup

WiFi FTMs. We setup our vehicle with a small form factor computer that contains two Intel Dual Band Wireless-AC 8260. Each WiFi transceiver connects to two WiFi external antennas: 6dBi RP-SMA Dual Band 2.4GHz 5GHz with 1.637 m cable to attach the antennas on the roof (we subtract that length, c from FTM ranges). We leverage an open Linux FTM tool [26] to initiate and extract FTMs from these WiFi cards. On the AP side, we use ASUS Wireless-AC1300 RT-ACRH13 APs which are configured to respond to FTM requests as a built-in capability.

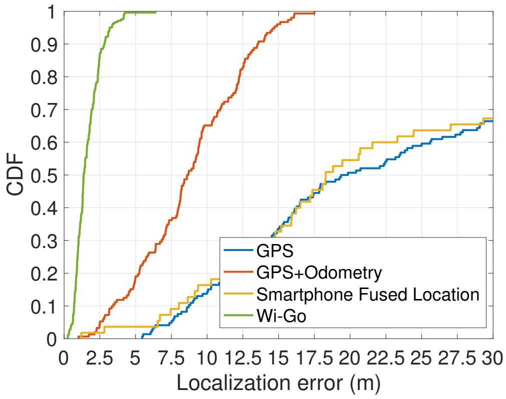
Vehicle Odometry. We connect OBDLink MX to the vehicle OBD-II interface, and retrieve odometry readings to the small form factor computer through Bluetooth interface (Plugable USB adapter).

4.7.2 Urban Canyon Evaluation

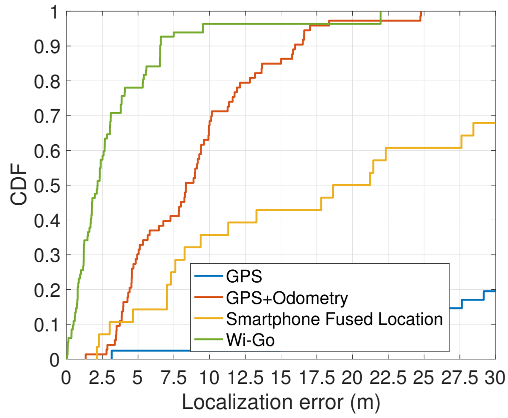
In this section, we evaluate Wi-Go in terms of localization error through an urban canyon experiment in which GPS is significantly degraded. Moreover, we show how Wi-Go will react in terms of latency, as many APs and vehicles are actively contending over WiFi channels through WiFi FTM packets and usual WiFi data traffic in urban canyon.

Experiment Summary. Fig.4.2(a) illustrates one of our experiments in upper Manhattan, New York City. We park four vehicles in a single street (182.6 m long) as shown in the figure as red stars, and we place an AP on each vehicle. In the fifth vehicle, we use our form factor PC configured as a WiFi station (STA), which continuously ranges surrounding APs. We also collect odometry, vehicle GPS, and standalone GPS readings from this vehicle. The fifth vehicle drives down the street multiple times to gain statistically meaningful results. We repeated the same experiment in Midtown Manhattan, where we placed six APs inside local shops.

Ground Truth. In obstructed environments like urban canyons, even high precision GPS will experience large errors. It is also infeasible to manually measure a vehicle's location while driving. Due to these reasons, getting accurate ground truth to compare our results



(a) Parked Cars (114th st.)



(b) Indoor APs (49th st.)

Figure 4.7: Vehicle tracking error of Wi-Go Vs current localization systems in Manhattan experiments.

Approach	Parked Cars	Indoor APs
Wi-Go	1.9 m	3.6 m
GPS+Odometry	14.3 m	16.4 m
GPS	18.9 m	27.4 m
Smartphone Fused Loc.	17.8 m	16 m

Table 4.2: AP localization mean error in Manhattan experiments.

to was challenging. We obtain ground truth by leveraging depth cameras as mentioned earlier in Sec. 4.3. To validate our ground truth methodology, we compare depth readings of a light pole from depth camera and laser range finder (or measuring tape) over a range of 3 to 12 m in a parking lot environment. The resulting error is within 0.73 m. To obtain the ground truth location of the APs, we take pictures of the APs' surrounding environments (buildings and other landmarks), cross reference with Google Street View, and drop pins accordingly on Google Maps to obtain the coordinates of the APs. Prior studies show that Google Earth has an accuracy close to 1 m in metropolitan cities like Montreal [118] and Rome [119].

For these experiments, we compare the performance of Wi-Go to other technologies mentioned earlier. In Fig. 4.7(a), we show the cumulative distribution function (CDF) of the localization error of the vehicle using Wi-Go. Our Wi-Go achieves 1.3 m median error, and 2.8 m 90-percentile. In contrast, vehicle GPS (GPS+Odometry) achieves 9.04 m median error, and 13.5 m 90-percentile, compared to standalone GPS achieving 19.6 m median error,

Source	Single FTM	Processing
Median Latency (ms)	20	2.2

Table 4.3: Wi-Go Latency.

and 42.1 m 90-percentile. Finally, Android Fused Location Provider API achieves 18.2 m median error, and 58.09 m 90-percentile. In another experiment (Fig. 4.7(b)), where APs were inside of local shops, the median tracking error of Wi-Go is 2.1 m, while 90-percentile error increases to 6.5 m due to extra signal degradation caused by concrete walls of these shops.

Table 4.2 shows the AP localization error of Wi-Go system when compared to baseline technology options. As the vehicle moves across the street with more rounds of driving, Wi-Go improves the AP localization accuracy, reaching 1.9 m mean localization error across all the APs. This result demonstrates a significant performance improvement of Wi-Go system over baseline solution. For the baseline achieving 14.3 m mean error, we utilize the traces via vehicle GPS to estimate the locations of APs using multilateration and FTM ranges.

Median latency is reported in Table 4.3. The median latency to acquire a single FTM reading, i.e. the time difference between the moment that we make a system call to initiate FTM process to the moment that call returns with range measurements from a single AP. We measure this median latency, in our current setup, to be 20 ms. On the other hand, the median processing latency of FTMSLAM is 2.2 ms. The total latency of Wi-Go system is controlled by the latency of extracting the sensing information such as WiFi FTM and OBD readings. With a normal vehicle speed, this measured latency may lead to error of a few meters. We believe that this latency could be improved in the actual production system by optimizing the FTM extraction tool.

Simulating Dense Environment. We use ns-3.30.1 to simulate a scenario of dense environment including 900 vehicles and 200 APs. The mobility trace is generated using SUMO mobility simulator [120] for an approximately 500 m radius around Times Square, NYC, imported from OpenStreetMap [121]. We imported a map of the selected neighborhood in Manhattan to create both the mobility traces of the vehicles and determine the location of buildings to increase simulation realism.

In this simulation, we implemented the WiFi FTM protocol following the technical specification of IEEE 802.11 standard [65] (e.g., packet format and packets size). For system parameters that are not directly regulated in the standard (e.g., expiration timers and other supporting parameters), we directly infer them from our empirical experiments. We log the latency for getting a single FTM ranging in our simulation.

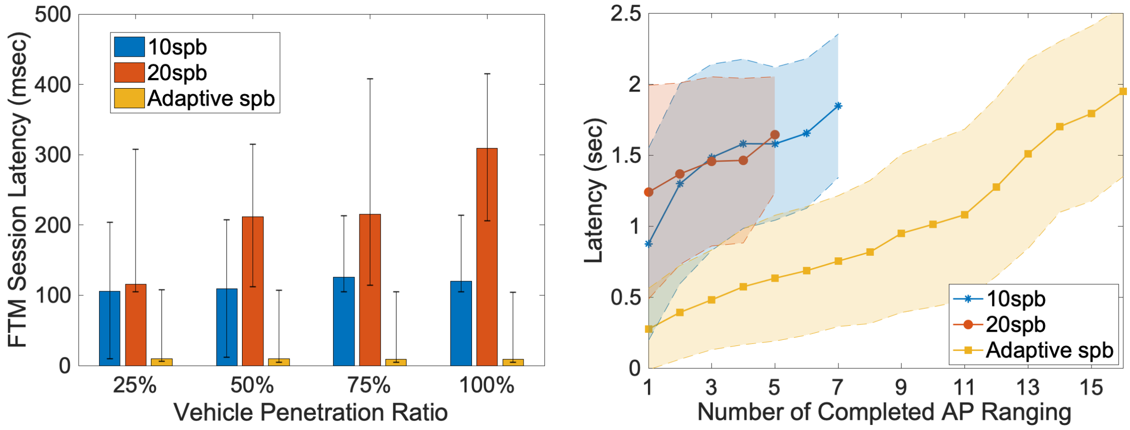
Simulation Parameter	**Value**
Simulation time	30 sec
Transmission power	16.5 dBm
Channel bandwidth	80 MHz
Channel number	155
Line-of-sight reference pathloss	21.87 dB
Line-of-sight pathloss exponent	3.39

Table 4.4: Simulation configuration.

Table 4.4 shows the configuration used for the simulation. To better capture the impact of the multipath effects in an urban environment, the simulator distinguishes obstructed, none-line-of-sight (NLOS) communications from line-of-sight (LOS) communication as follows: when there is a packet transmission, the simulator identifies the communication category and then chooses an appropriate propagation model to calculate the received signal at the receiver. The obstructed building signal propagation model is an implementation of Mangel et al. [122]. For the line-of-sight model we used pathloss components from a recent industry consortium (CAMP) [123] empirical measurement study, which considers the impact of vehicular traffic condition on the pathloss.

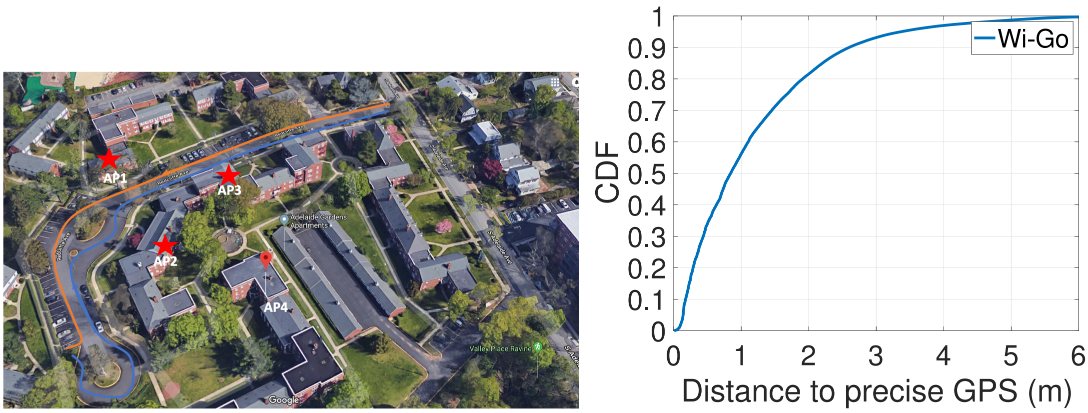
Fig. 4.8(a) illustrates the median latency of different approaches, grouped together for specific vehicle densities. The error bars indicate $25^{\text{th}}\%$ and $75^{\text{th}}\%$ of the latency for each bars. Vehicle density is captured by the percentage of cars equipped with Wi-Fi, which we call vehicle penetration ratio, and simulation results are presented for four penetration ratios. It is clearly shown that Wi-Go, labeled as *Adaptive spb*, significantly outperforms the other competing approaches.

Fig. 4.8(b) illustrates the end-to-end latency from the moment that a vehicle starts with a list of APs until it successfully finishes its entire FTM session with the n^{th} AP, averaged across entire simulation. Note that the vehicle starts over with a fresh list of scanned APs



(a) (b)

Figure 4.8: (a) $25^{th}\%$, median, and $75^{th}\%$ latency for Wi-Go (adaptive spb) compared to baselines using ns-3 simulations, and (b) mean and standard deviation of the latency versus number of completed AP rangings in one AP list scan for 75% penetration ratio.



(a) Residential Experiment (b) CDF

Figure 4.9: Evaluating vehicle tracking accuracy in the residential apartment complex experiment.

either after finishing the current list of APs or after 2.5 s, whichever comes first. The shaded area around each curve shows the standard deviation. It is shown that on average the vehicle can start triangulation by having the third FTM session completed after 500 msec with our proposed adaptive spb approach; in contrast, using 10spb and 20spb approaches, the vehicle has to wait approximately 1500 msec and cannot complete ranging with more than seven APs.

Approach	Mean	Min	Max
Wi-Go	2.6 m	1.3 m	4 m
GPS+Odometry	3.6 m	2.9 m	4.3 m

Table 4.5: AP localization error in the residential apartment complex experiment.

4.7.3 Residential Apartments Evaluation

Summary. In our second experiment, we evaluate Wi-Go in a residential apartment complex area. We place three APs inside student apartments, in the location where residents usually keep their access point, as shown in Fig. 4.9(a). The choice of apartment units was limited to a set of volunteers who provided access to their units. Our data collection has been scheduled over a week with two trips each day (morning and afternoon). One trip refers to each marked colored path in Fig. 4.9(a) (e.g. blue path is one trip).

Ground truth. Ground truth locations of APs are estimated by finding the latitude and longitude of the nearest window on Google Maps and correcting with manually measured AP distance to the window. Regarding vehicle localization error, we use high precision GPS as ground truth to evaluate vehicle localization error.

Fig. 4.9(b) shows the CDF of the vehicle localization error for this residential area. Wi-Go, with vehicle initialization through accurate GPS measurement, achieves 0.8 m median localization error, and 3.2 m 90-percentile. Table 4.5 summarizes the AP localization error for Wi-Go compared to the baseline solution (GPS corrected with odometry readings). Our Wi-Go can determine the locations of the APs with 2.6 m mean error over all APs compared to 3.6 m mean error for the baseline. For this baseline result, we filtered noisy GPS readings using our uncertainty weights for GPS, and then used FTM measurements and multilateration to determine the AP's position.

4.7.4 Micro Benchmarks

Effect of varying number of APs. The density of APs is an important factor affecting vehicle localization error. In Fig. 4.10, we study this effect in the Manhattan experiment, in which WiFi FTM dominantly affect localization error compared to suburban areas. This figure shows that, as expected, the median of vehicle localization error decreases with increasing number of APs, as we increase more reference points.

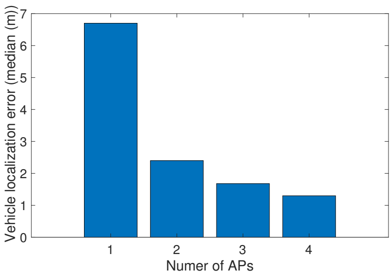


Figure 4.10: APs density effect on vehicle's localization error in Manhattan.

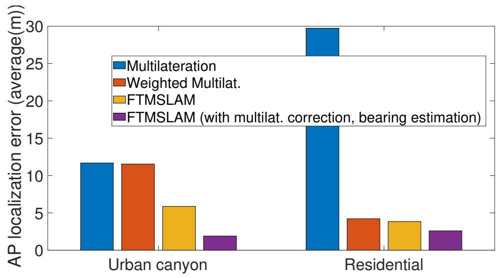


Figure 4.11: FTMSLAM algorithms effect on AP localization error.

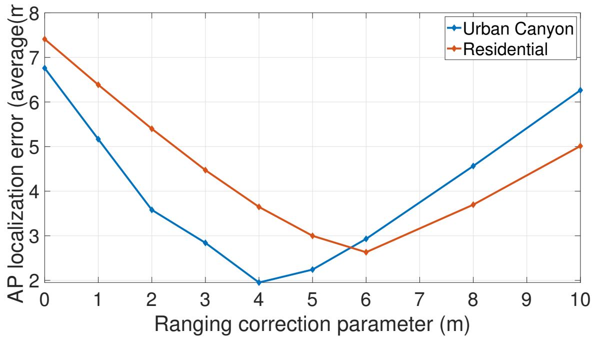
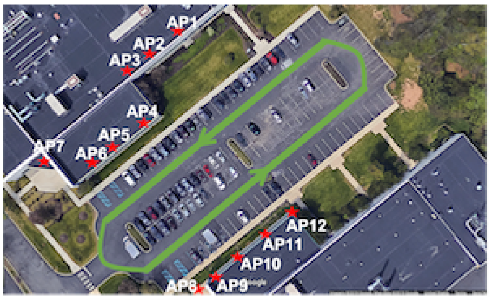


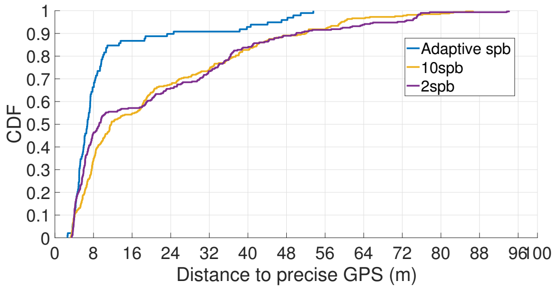
Figure 4.12: Ranging correction effect on AP localization error.

Effect of different FTMSLAM algorithms. We compare here the impact of our proposed algorithms over standard multilateration on AP localization error. Fig. 4.11 shows that our algorithms can gradually improve the average localization error of APs. In our urban canyon dataset with low-quality GPS measurements, adding our GPS and FTM weights to the standard multilateration does not improve the accuracy significantly. In contrast, as we apply FTMSLAM, our tracking framework with our crafted weights, the average localization error drops to $5.8m$. When we apply our complete FTMSLAM, by correcting vehicle's particles with FTM multilateration estimate and bearing estimation, the AP localization error is further reduced to 1.9 m. In the residential experiment, Fig. 4.11 shows that our GPS and FTM weights could significantly improve the average localization error of APs, since a number of outliers exist in these measurements. After we apply complete FTMSLAM, the AP localization error decreases from 4.2 m to 2.6 m.

Effect of varying FTM correction factor. FTM ranging is affected by multipath as shown in previous work [26]. We study how the correction factor, which we subtract from the FTM ranges, could affect FTMSLAM in terms of AP localization error. Fig. 4.12 indicates



(a) Scalability Experiment.



(b) CDF

Figure 4.13: Evaluating our adaptive FTM message rate approach leveraging FTM ranges only.

that there is an optimization value that minimizes the AP localization error. According to our evaluation, this value is different between our two datasets; as expected, it is higher in the residential dataset compared to the urban canyon dataset. This is because, in the residential scenarios, the APs are placed inside the building and thus more susceptible to multipath, in direct contrast to urban scenarios where APs are on the top of the vehicles characterized with Line-of-Sight signal propagation. Our FTMSLAM derives this FTM correction factor **automatically** through our adaptive FTM range correction algorithm.

AP Location Convergence. Wi-Go improves APs location estimations over time. Wi-Go can converge to meter-level accuracy of an AP location estimation with relatively small number of visits to this AP. Specifically, Wi-Go can converge in Manhattan to 1.9 m, 3.6 m average AP localization error after 10, 3 visits to parked cars, indoor APs (114th st., 49th st.) with 488, 3273 average FTM readings, respectively. The number of readings is a function of the number of visits and the traffic condition (average vehicle speed). Our system can also converge to 2.6 m AP localization error after 14 visits to a residential area with 150 average FTM readings. Note that as more vehicles visit an AP, AP location accuracy improves and, as a result, vehicle localization accuracy improves.

4.7.5 Evaluating Adaptive FTM Rate

In this section, we evaluate our solution to adapting the FTM message rate to maximize vehicle localization accuracy, while being constrained to a cumulative message rate.

Experimental Results. We conduct an experiment in a suburban area, in which we place 12 APs (first seven are indoors and the rest are outdoor) as shown in Fig. 4.13(a). In this experiment we aim to show the importance of wisely ranging to a subset of the nearby APs, as well as to study how this mechanism affects the vehicle localization accuracy. To do so, 12 APs are deployed in an area of 117m X 36.5m, and we assume that the locations of these APs are already estimated through previous steps (wardriving multilateration, or FTMSLAM with minimal sample per burst). To avoid any complication, we do not use any preprocessing algorithms (e.g., FTM uncertainty weighting or FTMSLAM), but we utilize nearby APs to track the vehicle using standard multilateration and count only on WiFi FTMs. We use a high precision GPS receiver as ground truth to calculate the final localization error.

Fig. 4.13(b) compares our adaptive approach to the default approach using fixed spb per AP (2 and 10). As expected, the 2*spb* approach is susceptible to FTM noise, resulting in a high localization error of 9.2 m median error, and more importantly, 50 m 90-percentile. On the other hand, by increasing the parameter to 10*spb*, it is expected to improve the location accuracy, as there are more samples to average out the noise. However, to our surprise, this higher spb approach generates 11.5 m median error, and 52.1 m 90-percentile. This counter-intuitive observation could be explained as follows: Though we increased the sampling set for statistical accuracy, we also significantly increase the opportunity to collect more inaccurate, noisy data from faraway APs.

Our proposed adaptive approach, instead, takes all these factors into account. Our adaptive algorithm achieves 6.7 m median error, and 24.6 m 90-percentile, respectively. These results validate that our approach improves the localization error, since it intelligently selects the subset of APs which are nearby, less affected by FTM noise, and with a lower horizontal dilution of precision.

4.7.6 Discussion

In this section we discuss the major lessons learnt from our study and the results obtained from different scenarios are related and scrutinized.

Vehicle location error vs. AP localization error. As shown in Fig. 4.7(a) and Table 4.2, we notice that the vehicle localization error is slightly lower than the AP localization error, which is attributed to two factors: first, the multilateration reference points (i.e., vehicle trace) used to locate the AP are collinear, while the reference points used to locate the vehicle (i.e., widely spread AP locations) are mostly not, leading to lower dilution of multilateration precision. Second, AP localization is only limited by FTM measurements; the constraints on vehicle location, on the other hand, come down to GPS, odometry, and FTM measurements. This results in more accurate location estimation.

Urban canyon results vs. residential results. The results using the urban canyon and residential datasets consistently show that FTMSLAM alleviates both GPS blockage in obstructed environments and occasional GPS noise in unobstructed environments. Interestingly, the average AP localization error is slightly higher in residential areas, due to signal multipath. We observe that it is imperative to initialize our system with sufficiently accurate location as initial conditions, which is possible using GPS in the residential area, or using Maps/FTMs in urban canyons. Wi-Go is designed to filter noisy measurements when it happens, thus only benefiting from reasonably accurate measurements. As shown in Fig. 4.11, the different components of our system have different impacts between the two datasets.

4.8 Related Work

Outdoor Localization. In urban canyons, GPS can achieve on average 24.3m error [75], which can be reduced by 6-8m as shown recently [124]. ParkLoc [90] localizes cars with an accuracy of 4.8m in underground parking garages using inertial sensors in the smartphones of the people inside the car, and semi-supervised learning algorithms. Carloc [75] can track vehicles outdoors (2.7 m mean error) by matching digital maps and leveraging proprietary vehicle sensors to detect roadway landmarks (e.g. speed bumps, stop signs). The LTE standard [125] indicates that a localization accuracy of 20-30m can be achieved with trilateration methods and using neural networks in tandem allows for a minimum error of 2.3m to be possible [126].

WiFi Localization. Indoor localization either assumes the knowledge of APs locations [7, 127] or leverage fingerprinting techniques [12, 128, 129]. Except for fingerprinting, the position of the AP is usually assumed to be known. Locating APs while driving by through ware-driving could lead to up to 32m error [130]. This can be improved by estimating angle of arrival reaching 10-30m localization error [131]. Fingerprinting strategies, however, involve heavy overheads, relatively low accuracy compared to ToA methods similar to GPS. CUPID [132] leverages CSI for indoor tracking, achieving 2.7m median error. WOLoc [92] offers WiFi RSSI-based outdoor tracking through semi-supervised manifold learning technique.

Scalability. Banin et al. [133] proposes a solution using multiple access points which are custom made for using WiFi FTM and communicating with each other to form a geosynchronous system, where the clients track their positions passively, with only the APs transmitting. Llombart et al. in [134] simulate a technique that can be used for mobile devices to select three nearby APs for trilateration. The mobile device to be positioned finds three nearby APs, associates with each of the three APs, calculates the distance and then dissociates. Dedicated hardware is required for both the AP and the mobile device in this method.

4.9 Conclusion

This work introduces Wi-Go, a vehicle localization system that uses WiFi FTM measurements to achieve lane-level accuracy in challenging urban canyons where GPS accuracy is degraded. It fuses WiFi FTMs with GPS and odometry in the FTMSLAM framework, to simultaneously estimate vehicle positions and the positions surrounding WiFi APs. We further design Wi-Go to adapt the rate of FTM packets in order to mitigate network congestion and latency arising from WiFi FTM's active ranging approach, while maximizing vehicle localization accuracy. We evaluate the system in the urban canyons of Manhattan as well as suburban residential areas. Wi-Go achieves 1.3 m median error in an urban canyon setting with access points on parked vehicles, 2.1m median error in a crowded area with access points in buildings, and 0.8 m median localization error in a suburban environment with access points inside apartment buildings. This shows promise for using WiFi FTM measurements for vehicle positioning even in multi-path rich urban canyons.

Chapter 5

Conclusion

This dissertation introduces accurate indoor and outdoor tracking that can adaptively learn the environment and map anchor nodes. We leverage visible light for indoor device-free passive tracking and WiFi Fine Time measurements for outdoor vehicle tracking. We pick these unconventional wireless signals, as their infrastructure are ubiquitously deployed, and can accurately detect presence and provide range measurements. We design our tracking systems not only to be accurate, but also to scale with many users and can be easily deployed. We design, build, prototype and evaluate these tracking systems through real word experiments.

In summary, this thesis details the following contributions:

- EyeLight, a device-free tracking system embedded in indoor lighting environment to detect human presence, estimate occupancy and room activities. This system envisions smart light bulbs that can sense the light reflected off the floor. We design localization algorithms that automatically estimate variations in light level and detect human presence, track users, estimate room occupancy and recognize activities. This system achieves submeter-level localization accuracy in a conference room.

- An experimental framework to evaluate time-based ranging systems; an open platform for experimenting with WiFi Fine Time Measurements. Our experiments confirm meter-level ranging accuracy is possible as promised, but the measurements also show that this can only be consistently achieved in low-multipath environments such as open outdoor spaces.

- Wi-Go, a vehicle positioning system that coherently fuses WiFi Fine Time Measurements with GPS and odometery information through FTMSLAM framework. FTMSLAM can simultaneously track vehicles and map WiFi access points through vehicles crowd-sourcing. Wi-Go can adapt the rate of FTM packets in order to mitigate network congestion and latency arising from WiFi FTM's active ranging. Wi-Go achieves meter-level localization in multi-path rich urban canyons such as in Manhattan, New York City.

5.1 Lessons Learned

Our results, in this dissertation, confirm the following:

- Submeter device-free tracking accuracy is achievable indoor using visible light. Moreover, coarse grained occupancy estimation, and activity detection can be built over such accurate tracking. Light bulbs can be mapped using either LiDAR assisted robot or crowd-sourced through users if map of ceiling power outlets are not available.

- Meter-level indoor tracking using only WiFi Fine Time Measurements, given their off-the-shelf APs location, is still a challenge. However, we envision that fusing WiFi FTM with other modalities like motion sensors and Bluetooth can improve such accuracy. Moreover, standardizing WiFi FTM in the 60GHz band can improve such tracking accuracy while may limit its range.

- Lane-level outdoor positioning is possible through a careful fusion of WiFi FTM, GPS, and vehicle odometry readings while simultaneously mapping APs location. Improving such accuracy, requires incorporating street maps and utilizing denser deployment of WiFi access points supporting FTM.

5.2 Future Directions

Looking forward, WiFi Fine Time Measurements can enable many tracking applications. With incorporating WiFi FTM in the 60 GHz signals that can improve range error to tens of centimeters, wider range of tracking applications can be achieved. There are several directions to build on this thesis work:

- An interesting extension to WiFi FTM is to estimate angle-of-arrival along with the ranging measurement that could be through CSI information and antenna array, or through simulating antenna array using inverse synthetic aperture approach.
- Device-free tracking using WiFi FTM and antenna array can also be explored.
- Currently, smart phones can estimate pairwise distance to neighboring phones through WiFi FTM. These measurements can enable accurate alarm system for social distancing and contact tracing.

References

[1] R. Want, A. Hopper, V. Falcão, and J. Gibbons, “The active badge location system,” *ACM Trans. Inf. Syst.*, vol. 10, p. 91102, Jan. 1992.

[2] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster, “The anatomy of a context-aware application,” in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom ’99, (New York, NY, USA), p. 5968, Association for Computing Machinery, 1999.

[3] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The cricket location-support system,” in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom ’00, (New York, NY, USA), p. 3243, Association for Computing Machinery, 2000.

[4] J. Xiong and K. Jamieson, “Arraytrack: A fine-grained indoor location system,” in *Presented as part of the 10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13)*, pp. 71–84, 2013.

[5] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, “Spotfi: Decimeter level localization using wifi,” in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pp. 269–282, 2015.

[6] J. Xiong, K. Sundaresan, and K. Jamieson, “Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization,” in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, MobiCom ’15, (New York, NY, USA), pp. 537–549, ACM, 2015.

[7] D. Vasisht, S. Kumar, and D. Katabi, “Decimeter-level localization with a single wifi access point,” in *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, pp. 165–178, 2016.

[8] P. Bahl and V. N. Padmanabhan, “Radar: an in-building rf-based user location and tracking system,” in *Proc. IEEE INFOCOM 2000 Conf. on Comput. Commun. 19th Annu. Joint Conf. IEEE Comput. and Commun. Soc.*, vol. 2, pp. 775–784 vol.2, 2000.

[9] T. Roos, P. Myllymäki, H. Tirri, P. Misikangas, and J. Sievänen, “A probabilistic approach to wlan user location estimation,” *International Journal of Wireless Information Networks*, vol. 9, no. 3, pp. 155–164, 2002.

[10] M. Youssef and A. Agrawala, “The horus wlan location determination system,” in *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, MobiSys ’05, (New York, NY, USA), p. 205218, Association for Computing Machinery, 2005.

[11] R. Ayyalasomayajula, A. Arun, C. Wu, S. Sharma, A. R. Sethi, D. Vasisht, and D. Bharadia, "Deep learning based wireless localization for indoor navigation," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, MobiCom '20, (New York, NY, USA), Association for Computing Machinery, 2020.

[12] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, *et al.*, "Place lab: Device positioning using radio beacons in the wild," in *International Conference on Pervasive Computing*, pp. 116–133, Springer, 2005.

[13] M. Ibrahim and M. Youssef, "Cellsense: A probabilistic rssi-based GSM positioning system," in *IEEE Globecom*, 2010.

[14] M. Youssef, M. Mah, and A. Agrawala, "Challenges: Device-free passive localization for wireless environments," in *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, MobiCom '07, (New York, NY, USA), p. 222229, Association for Computing Machinery, 2007.

[15] W. Brendel, M. Amer, and S. Todorovic, "Multiobject tracking as maximum weight independent set," in *CVPR 2011*, pp. 1273–1280, 2011.

[16] S. Bae and K. Yoon, "Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1218–1225, 2014.

[17] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, "Mots: Multi-object tracking and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7942–7951, 2019.

[18] T. S. Ralston, G. L. Charvat, and J. E. Peabody, "Real-time through-wall imaging using an ultrawideband multiple-input multiple-output (mimo) phased array radar system," in *2010 IEEE International Symposium on Phased Array Systems and Technology*, pp. 551–558, 2010.

[19] K. Chetty, G. E. Smith, and K. Woodbridge, "Through-the-wall sensing of personnel using passive bistatic wifi radar at standoff distances," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 4, pp. 1218–1226, 2012.

[20] F. Adib and D. Katabi, "See through walls with wifi!," in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pp. 75–86, 2013.

[21] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3d tracking via body radio reflections," in *Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation*, NSDI'14, (USA), p. 317329, USENIX Association, 2014.

[22] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures," in *Proc. of the 20th Annual Int. Conf. on Mobile Computing and Networking*, pp. 617–628, ACM, 2014.

[23] T. Li, Q. Liu, and X. Zhou, "Practical human sensing in the light," in *Proc. of the 14th Annu. Int. Conf. on Mobile Syst., Applicat., and Services*, MobiSys '16, (New York, NY, USA), pp. 71–84, ACM, 2016.

[24] R. Ayyalasomayajula, A. Arun, C. Wu, A. Shaikh, S. Rajagopalan, Y. Hu, S. Ganesaraman, C. J. Rossbach, A. Seetharaman, E. Witchel, *et al.*, "Locap: Autonomous millimeter accurate mapping of wifi infrastructure," in *17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 20)*, pp. 1115–1129, 2020.

[25] V. Nguyen, M. Ibrahim, S. Rupavatharam, M. Jawahar, M. Gruteser, and R. Howard, "Eyelight: Light-and-shadow-based occupancy estimation and room activity recognition," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pp. 351–359, 2018.

[26] M. Ibrahim, H. Liu, M. Jawahar, V. Nguyen, M. Gruteser, R. Howard, B. Yu, and F. Bai, "Verification: Accuracy evaluation of wifi fine time measurements on an open platform," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pp. 417–427, ACM, 2018.

[27] M. Ibrahim, A. Rostami, B. Yu, H. Liu, M. Jawahar, V. Nguyen, M. Gruteser, F. Bai, and R. Howard, "Wi-go: accurate and scalable vehicle positioning using wifi fine timing measurement," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pp. 312–324, 2020.

[28] J. Yun and S.-S. Lee, "Human movement detection and identification using pyroelectric infrared sensors," *Sensors*, vol. 14, no. 5, pp. 8057–8081, 2014.

[29] C. R. Wren and E. M. Tapia, "Toward scalable activity recognition for sensor networks," in *LoCA*, vol. 3987, pp. 168–185, 2006.

[30] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and co2 measurements using statistical learning models," *Energy and Buildings*, vol. 112, pp. 28–39, 2016.

[31] Z. Chen, R. Zhao, Q. Zhu, M. K. Masood, Y. C. Soh, and K. Mao, "Building occupancy estimation with environmental sensors via cdblstm," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 12, pp. 9549–9559, 2017.

[32] J. Lei, X. Ren, and D. Fox, "Fine-grained kitchen activity recognition using rgb-d," UbiComp '12, pp. 208–211, ACM, 2012.

[33] M. Keally, G. Zhou, G. Xing, J. Wu, and A. Pyles, "Pbn: Towards practical activity recognition using smartphone-based body sensor networks," in *Proc. of the 9th ACM Conf. on Embedded Networked Sensor Syst.*, pp. 246–259, ACM, 2011.

[34] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, "Epsilon: A visible light based positioning system," in *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, (Seattle, WA).

[35] C. Zhang and X. Zhang, "Litell: Robust indoor localization using unmodified light fixtures," in *Proc. of the 22nd Annu. Int.Conf. on Mobile Computing and Networking*, MobiCom '16, (New York, NY, USA), pp. 230–242, ACM, 2016.

[36] T. Li, C. An, Z. Tian, A. T. Campbell, and X. Zhou, "Human sensing using visible light communication," in *Proc. of the 21st Annu. Int. Conf. on Mobile Computing and Networking*, MobiCom '15, (New York, NY, USA), 2015.

[37] Y. Yang, J. Hao, J. Luo, and S. J. Pan, "Ceilingsee: Device-free occupancy inference through lighting infrastructure based led sensing," in *Proc. of the 15th Int. Conf. on Pervasive Computing and Commun.*, Percom '17, IEEE, 2016.

[38] M. Ibrahim, V. Nguyen, S. Rupavatharam, M. Jawahar, M. Gruteser, and R. Howard, "Visible light based activity sensing using ceiling photosensors," in *Proc. of the 3rd Workshop on Visible Light Commun. Syst.*, VLCS '16, (New York, NY, USA), pp. 43–48, ACM, 2016.

[39] `http://www.energy.gov/energysaver/led-lighting`.

[40] `http://www.ledsmagazine.com/articles/2005/01/benefits-and-drawbacks-of-leds.html`.

[41] D. Kelly, "Sine waves and flicker fusion," *Documenta Ophthalmologica*, vol. 18, no. 1, pp. 16–35, 1964.

[42] M. Ibrahim, M.and Youssef, "Cellsense: An accurate energy-efficient gsm positioning system," *IEEE Trans. on Vehicular Technology*, 2012.

[43] Z. K. F. Adib and D. Katabi, "Multi-person localization via rf body reflections," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI'15)*, (Oakland, CA).

[44] K. E. Caine, A. D. Fisk, and W. A. Rogers, "Benefits and privacy concerns of a home equipped with a visual sensing system: A perspective from older adults," in *Proc. of the human factors and ergonomics society annual meeting*, vol. 50, 2006.

[45] M. Valtonen, J. Maentausta, and J. Vanhala, "Tiletrack: Capacitive human tracking using floor tiles," in *2009 IEEE Int. Conf. on Pervasive Computing and Commun.*, pp. 1–10, March 2009.

[46] R. J. Orr and G. D. Abowd, "The smart floor: A mechanism for natural user identification and tracking," in *CHI '00 Extended Abstracts on Human Factors in Computing Syst.*, CHI EA '00, (New York, NY, USA), pp. 275–276, ACM, 2000.

[47] `http://ab.rockwellautomation.com/Sensors-Switches/Operator-Safety/Light-Curtain`.

[48] `https://www.pepperl-fuchs.com/global/en/classid_4294.htm`.

[49] P. Nguyen, U. Muncuk, A. Ashok, K. R. Chowdhury, M. Gruteser, and T. Vu, "Battery-free identification token for touch sensing devices," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, SenSys '16, (New York, NY, USA), pp. 109–122, ACM, 2016.

[50] Z. Li, W. Chen, C. Li, M. Li, X.-Y. Li, and Y. Liu, "Flight: Clock calibration using fluorescent lighting," in *Proc. of the 18th Annu. Int. Conf. on Mobile Computing and Networking*, Mobicom '12, (New York, NY, USA), pp. 329–340, ACM, 2012.

[51] B. J. Mohler, W. B. Thompson, S. H. Creem-Regehr, H. L. Pick, and W. H. Warren, "Visual flow influences gait transition speed and preferred walking speed," *Experimental brain research*, vol. 181, no. 2, pp. 221–228, 2007.

[52] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *European conference on computational learning theory*, pp. 23–37, Springer, 1995.

[53] `http://www.ti.com/product/LF356`.

[54] `http://www.ti.com/lit/ds/symlink/ina126.pdf`.

[55] `http://www.ti.com/product/MSP432P401R`.

[56] `http://www.ti.com/tool/cc3100boost`.

[57] `https://www.stereolabs.com/`.

[58] X. Li, K. Pahlavan, M. Latva-aho, and M. Ylianttila, "Comparison of indoor geolocation methods in dsss and ofdm wireless lan systems," in *Vehicular Technology Conference Fall 2000. IEEE VTS Fall VTC2000. 52nd Vehicular Technology Conference (Cat. No.00CH37152)*, vol. 6, pp. 3015–3020 vol.6, 2000.

[59] D. D. McCrady, L. Doyle, H. Forstrom, T. Dempsey, and M. Martorana, "Mobile ranging using low-accuracy clocks," *IEEE Transactions on Microwave Theory and Techniques*, vol. 48, no. 6, pp. 951–958, 2000.

[60] A. Günther and C. Hoene, "Measuring round trip times to determine the distance between wlan nodes," in *International Conference on Research in Networking*, pp. 768–779, Springer, 2005.

[61] M. Ciurana, F. Barcelo-Arroyo, and F. Izquierdo, "A ranging system with ieee 802.11 data frames," in *Radio and Wireless Symposium, 2007 IEEE*, pp. 133–136, IEEE, 2007.

[62] S. A. Golden and S. S. Bateman, "Sensor measurements for wi-fi location with emphasis on time-of-arrival ranging," *IEEE Transactions on Mobile Computing*, vol. 6, no. 10, 2007.

[63] D. Giustiniano and S. Mangold, "Caesar: carrier sense-based ranging in off-the-shelf 802.11 wireless lan," in *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, p. 10, ACM, 2011.

[64] S. Lanzisera, D. T. Lin, and K. S. Pister, "Rf time of flight ranging for wireless sensor network localization," in *Intelligent Solutions in Embedded Systems, 2006 International Workshop on*, pp. 1–12, IEEE, 2006.

[65] ""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.

[66] https://goo.gl/BSUCdG. Wi-Fi CERTIFIED Location.

[67] https://goo.gl/1z1LhE. Android Pie Indoor Positioning.

[68] L. Banin, U. Schatzberg, and Y. Amizur, "Wifi ftm and map information fusion for accurate positioning," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2016.

[69] X. Li and K. Pahlavan, "Super-resolution toa estimation with diversity for indoor geolocation," *IEEE Transactions on Wireless Communications*, vol. 3, pp. 224–234, Jan 2004.

[70] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[71] https://goo.gl/hQUyfo. Linux Core Releases.

[72] https://goo.gl/TzJRGG. FTM Patch for iw.

[73] M. Rea, A. Fakhreddine, D. Giustiniano, and V. Lenders, "Filtering noisy 802.11 time-of-flight ranging measurements from commoditized wifi radios," *IEEE/ACM Transactions on Networking*, vol. 25, pp. 2514–2527, Aug 2017.

[74] H. Wang, S. Sen, A. Mariakakis, R. Roy Choudhury, A. Elgohary, M. Farid, and M. Youssef, "Unsupervised indoor localization," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pp. 499–500, ACM, 2012.

[75] Y. Jiang, H. Qiu, M. McCartney, G. Sukhatme, M. Gruteser, F. Bai, D. Grimm, and R. Govindan, "Carloc: Precise positioning of automobiles," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pp. 253–265, ACM, 2015.

[76] R. Schmidt, "Least squares range difference location," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 32, pp. 234–242, Jan 1996.

[77] F. Izquierdo, M. Ciurana, F. Barceló, J. Paradells, and E. Zola, "Performance evaluation of a toa-based trilateration method to locate terminals in wlan," in *Wireless Pervasive Computing, 2006 1st International Symposium on*, pp. 1–6, IEEE, 2006.

[78] K. Saito, J.-I. Takada, and M. Kim, "Characteristics evaluation of dense multipath component in 11ghz-band indoor environment," in *Antennas and Propagation (EuCAP), 2016 10th European Conference on*, pp. 1–3, IEEE, 2016.

[79] A. Richter, "Estimation of radio channel parameters: Models and algorithms," ISLE, 2005.

[80] A. Mrstik and P. Smith, "Multipath limitations on low-angle radar tracking," *IEEE transactions on aerospace and electronic systems*, no. 1, pp. 85–102, 1978.

[81] M. Golmohamadi, S. Chowdhury, and J. Frolik, "Markov modeling of spatial variations in multipath," in *Antennas and Propagation & USNC/URSI National Radio Science Meeting, 2017 IEEE International Symposium on*, pp. 611–612, IEEE, 2017.

[82] D. E. Gustafson, J. R. Dowdle, J. M. Elwell, and K. W. Flueckiger, "A nonlinear code tracking filter for gps-based navigation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 4, pp. 627–638, 2009.

[83] J. A. Scheer, "Coherent radar system performance estimation," in *Radar Conference, 1990., Record of the IEEE 1990 International*, pp. 125–128, IEEE, 1990.

[84] M. May, E. Kreisher, T. Nasuti, and C. Sives, "Evaluation of gps receiver ranging accuracy," in *Position Location and Navigation Symposium, 1990. Record. The 1990's-A Decade of Excellence in the Navigation Sciences. IEEE PLANS'90., IEEE*, pp. 314–321, IEEE, 1990.

[85] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single wifi access point.," in *NSDI*, vol. 16, pp. 165–178, 2016.

[86] https://goo.gl/AW4Pto. Self-Driving Cars Market Growth.

[87] https://waymo.com/. Google WAYMO Self-Driving Car.

[88] C. Bo, X.-Y. Li, T. Jung, X. Mao, Y. Tao, and L. Yao, "Smartloc: Push the limit of the inertial sensor based metropolitan localization using smartphone," in *Proceedings of the 19th annual international conference on Mobile computing & networking*, pp. 195–198, ACM, 2013.

[89] H. Aly and M. Youssef, "Dejavu: an accurate energy-efficient outdoor localization system," in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 154–163, ACM, 2013.

[90] J. Cherian, J. Luo, and S.-S. Ho, "Parkloc: Light-weight graph-based vehicular localization in parking garages," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, p. 99, 2018.

[91] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm, "Accuracy characterization for metropolitan-scale wi-fi localization," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, pp. 233–245, ACM, 2005.

[92] J. Wang, N. Tan, J. Luo, and S. J. Pan, "Woloc: Wifi-only outdoor localization using crowdsensed hotspot labels," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9, IEEE, 2017.

[93] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *The international journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[94] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal, "Efficient, generalized indoor wifi graphslam," in *2011 IEEE International Conference on Robotics and Automation*, pp. 1038–1043, IEEE, 2011.

[95] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: Unsupervised indoor localization," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pp. 197–210, ACM, 2012.

[96] D. Giustiniano and S. Mangold, "Caesar: carrier sense-based ranging in off-the-shelf 802.11 wireless lan," in *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, p. 10, ACM, 2011.

[97] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *ACM SIGCOMM CCR*, vol. 41, p. 53, Jan. 2011.

[98] C. Pei, Z. Wang, Y. Zhao, Z. Wang, Y. Meng, D. Pei, Y. Peng, W. Tang, and X. Qu, "Why it takes so long to connect to a wifi access point," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, May 2017.

[99] P. Moutarlier and R. Chatila, "An experimental system for incremental environment modelling by an autonomous mobile robot," in *Experimental Robotics I*, pp. 327–346, Springer, 1990.

[100] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association," *Journal of Machine Learning Research*, vol. 4, no. 3, pp. 380–407, 2004.

[101] D. Hahnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose, "Mapping and localization with rfid technology," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 1, pp. 1015–1020, IEEE, 2004.

[102] E. Olson, J. J. Leonard, and S. Teller, "Robust range-only beacon localization," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 4, pp. 949–958, 2006.

[103] J. Djugash, S. Singh, G. Kantor, and W. Zhang, "Range-only slam for robots operating cooperatively with sensor networks," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pp. 2078–2084, IEEE, 2006.

[104] J. Djugash, S. Singh, and P. Corke, "Further results with localization and mapping using range from radio," in *Field and Service Robotics*, pp. 231–242, Springer, 2006.

[105] P. Newman and J. Leonard, "Pure range-only sub-sea slam," in *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 2, pp. 1921–1926, Ieee, 2003.

[106] J.-L. Blanco, J. González, and J.-A. Fernández-Madrigal, "A pure probabilistic approach to range-only slam," in *2008 IEEE International Conference on Robotics and Automation*, pp. 1436–1441, IEEE, 2008.

[107] B. Ferris, D. Fox, and N. D. Lawrence, "Wifi-slam using gaussian process latent variable models.," in *IJCAI*, vol. 7, pp. 2480–2485, 2007.

[108] R. Faragher, C. Sarno, and M. Newman, "Opportunistic radio slam for indoor navigation using smartphone sensors," in *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, pp. 120–128, IEEE, 2012.

[109] R. Berkvens, A. Jacobson, M. Milford, H. Peremans, and M. Weyn, "Biologically inspired slam using wi-fi," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1804–1811, IEEE, 2014.

[110] Y. Jiang, H. Qiu, M. McCartney, W. G. Halfond, F. Bai, D. Grimm, and R. Govindan, "Carlog: A platform for flexible and efficient automotive sensing," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pp. 221–235, ACM, 2014.

[111] G. Lui, T. Gallagher, B. Li, A. G. Dempster, and C. Rizos, "Differences in rssi readings made by different wi-fi chipsets: A limitation of wlan localization," in *2011 International conference on localization and GNSS (ICL-GNSS)*, pp. 53–57, IEEE, 2011.

[112] S. He and S. . G. Chan, "Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 466–490, 2016.

[113] R. B. Langley *et al.*, "Dilution of precision," *GPS world*, vol. 10, no. 5, pp. 52–59, 1999.

[114] G. Bansal, J. B. Kenney, and C. E. Rohrs, "Limeric: A linear adaptive message rate algorithm for dsrc congestion control," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4182–4197, 2013.

[115] G. Bansal, H. Lu, J. B. Kenney, and C. Poellabauer, "Embarc: Error model based adaptive rate control for vehicle-to-vehicle communications," in *Proceeding of the tenth ACM international workshop on Vehicular inter-networking, systems, and applications*, pp. 41–50, ACM, 2013.

[116] G. Bansal, B. Cheng, A. Rostami, K. Sjoberg, J. B. Kenney, and M. Gruteser, "Comparing limeric and dcc approaches for vanet channel congestion control," in *2014 IEEE 6th International Symposium on Wireless Vehicular Communications (WiVeC 2014)*, pp. 1–7, IEEE, 2014.

[117] T. Vincenty, "Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations," *Survey review*, vol. 23, no. 176, pp. 88–93, 1975.

[118] M. A. Goudarzi and R. J. Landry, "Assessing horizontal positional accuracy of google earth imagery in the city of montreal, canada," *Geodesy and Cartography*, vol. 43, no. 2, pp. 56–65, 2017.

[119] G. Pulighe, V. Baiocchi, and F. Lupia, "Horizontal accuracy assessment of very high resolution google earth images in the city of rome, italy," *International Journal of Digital Earth*, vol. 9, no. 4, pp. 342–362, 2016.

[120] https://sumo.dlr.de/. Simulation of Urban MObility (SUMO).

[121] https://www.openstreetmap.org/. Open Street Map.

[122] T. Mangel, O. Klemp, and H. Hartenstein, "5.9 ghz inter-vehicle communication at intersections: a validated non-line-of-sight path-loss and fading model," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, no. 1, p. 182, 2011.

[123] https://www.campllc.org/. Crash Avoidance Metrics Partners LLC (CAMP).

[124] X. Liu, S. Nath, and R. Govindan, "Gnome: A practical approach to nlos mitigation for gps positioning in smartphones," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 163–177, ACM, 2018.

[125] S. Chairman, "Summary of ad-hoc session on study on indoor positioning enhancements for utra and lte," in *3GPP TSG RAN WG1 Meeting*, pp. 20–24.

[126] Z. Shi and Y. Wang, "Neural network based localization using outdoor lte measurements," in *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, IEEE, 2018.

[127] A. T. Mariakakis, S. Sen, J. Lee, and K.-H. Kim, "Sail: Single access point-based indoor localization," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pp. 315–328, ACM, 2014.

[128] Q. D. Vo and P. De, "A survey of fingerprint-based outdoor localization," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 491–506, 2016.

[129] J. Rekimoto, "Placeengine: A wifi location platform based on realworld-folksonomy," in *Internet Conference 2006*, pp. 95–104, 2006.

[130] M. Kim, J. J. Fielding, and D. Kotz, "Risks of using ap locations discovered through war driving," in *International Conference on Pervasive Computing*, pp. 67–82, Springer, 2006.

[131] A. P. Subramanian, P. Deshpande, J. Gao, and S. R. Das, "Drive-by localization of roadside wifi networks," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pp. 718–725, IEEE, 2008.

[132] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding wifi localization," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pp. 249–262, ACM, 2013.

[133] L. Banin, O. Bar-Shalom, N. Dvorecki, and Y. Amizur, "Scalable wi-fi client self-positioning using cooperative ftm-sensors," *IEEE Transactions on Instrumentation and Measurement*, 2018.

[134] M. Llombart, M. Ciurana, and F. Barcelo-Arroyo, "On the scalability of a novel wlan positioning system based on time of arrival measurements," in *2008 5th Workshop on Positioning, Navigation and Communication*, pp. 15–21, IEEE, 2008.