UNIVERSITY OF CALIFORNIA
SANTA CRUZ

SELF-LOCALIZATION AND INTERPERSONAL PROXIMITY DETECTION FOR PUBLIC TRANSIT ACCESSIBILITY AND SAFETY

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE AND ENGINEERING

by

Fatemeh Mirzaei

March 2022

The Dissertation of Fatemeh Mirzaei
is approved:

Dr. Roberto Manduchi, Chair

Dr. Sri Kurniawan

Dr. Chen Qian

Dr. Mircea Teodorescu

Peter F. Biehl
Vice Provost and Dean of Graduate Studies

Table of Contents

Abstract

Self-Localization and Interpersonal Proximity Detection for Public Transit Accessibility and Safety

by

Fatemeh Mirzaei

Public transit stations and hubs are difficult to navigate for people with visual impairments. Moreover, public transit has been affected disproportionately by the social distancing requirements consequent to the COVID-19 pandemic. It is the objective of this dissertation to provide a technology for addressing these concerns in the frame of a mobile app named RouteMe2. The technology provides micro-routing and guidance to visually impaired travelers through complex routes in transit hubs. This work also includes the study to monitor the distance between the travelers inside the bus for social distancing application. Reducing the risk of air-born viral infections by social distancing can contribute to improving the overall safety of the public transit.

The key enablers of this technology are sufficiently accurate self-localization and micro-routing as well as effective communication of the contextual spatiotemporal information with the visually impaired users. The accuracy of the self-localization in the outdoor environments is challenged by poor Global Positioning System (GPS) reception due to tall nearby buildings that may obscure view of one or more satellites — a.k.a shading. Shading is very common in urban environments, and is a major cause of GPS failure. In order to mitigate the effect of shading, I statistically fuse the signals received from GPS as well as a small number of Bluetooth Low Energy (BLE) beacons. I further pair the statistical fusion with a Bayes discrete filter tracker to increase the self-localization accuracy. Experiments were conducted at San Jose Diridon light rail station to quantitatively assess the performance of the resulting system.

I have designed and implemented certain features and functionalities of RouteMe2 to provide effective communication of the in-context spatio-temporal information with visually impaired users while they use the app. I leveraged our previously published focus group study conducted with visually impaired people as well as reviewing the user interface of the existing related apps to improve the user experience of RouteMe2 the detail of which is presented.

I further assess the ability of two RSSI-based methods at detecting interpersonal distances shorter than 1 or 2 meters. One method uses the power received from the smartphone carried by another person. The other method measures the disparity in the power received by the two smartphones from one or more fixed BLE beacons. The results show that use of the RSSI disparity enables discrimination measures that are as good or better than using the RSSI received from another smartphone. I demonstrate the potential of a system that uses BLE beacons, placed inside a vehicle, to localize a passenger within the length of the vehicle with an accuracy better than 1 meter.

Acknowledgments

I would like to gratefully and sincerely thank my advisor, Dr. Roberto Manduchi, for his support, guidance, and patience. Beside all the knowledge I learned from him, his focus, patience, and objective questions helped me to be a better critical thinker, team worker, and a better problem solver. This work would not have been possible without his support and supervision.

Many thanks to my committee members Dr. Sri Kurniawan, Dr. Chen Qian, and Dr. Mircea Teodorescu for reading my dissertation and their advice. I learned invaluable lessons from Dr. Kurniawan classes that I have applied in this work.

To my lovely husband Amin Almassian, who always inspired me and for his constant encouragement and help during my study. His entrepreneurship mindset helped me to think out-of-the-box and see possibilities in challenging moments in my study. Thank you for helping me to become a better version of myself. Love you and could not have done it without all your support and love.

Special recognition goes out to my mom Malakeh Ajorloo and my dad Anamolah Mirzaei for their patience and encouragement throughout my study abroad. The text of this dissertation includes reprints of the following previously published material:

- Fatemeh Mirzaei, Roberto Manduchi, and Sri Kurniawan, "Public transit accessibility: Blind passengers speak out". In International Conference on Computers Helping People with Special Needs. Springer, 2018, pp. 277-282

- Fatemeh Mirzaei, Jonathan Lam, and Roberto Manduchi, "Accurate Self-Localization in Transit Stations: A Case Study" . In International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops). IEEE, 2020, pp. 1-6.

- Fatemeh Mirzaei and Roberto Manduchi, “Interpersonal Proximity Detection Using RSSI–Based Techniques. In Indoor Positioning and Indoor Navigation”. IPIN 2021.

- Fatemeh Mirzaei and Roberto Manduchi, “In–Vehicle Positioning for Public Transit Using BLE Beacons”.IPIN, 2021.

The co-author listed in this publication directed and supervised the research which forms the basis for the dissertation.

List of Figures

List of Tables

Chapter 1

Introduction

Public transit, when available, represents a safe, economical, and environmentally responsible way to travel. It is also the preferable, and sometimes the only, means of transportation for those who cannot drive, and who cannot rely on family or friends to be driven to places. These include elderly people or those who have lost their driver license due to medical reasons; people who are blind or have severe vision conditions (e.g. low acuity and tunnel vision); and people with a certain level of cognitive impairment (e.g. early stages of dementia). Although private transportation, in the form of taxi cabs or ride hailing services such as Uber and Lyft, is often available at least in urban areas, they may be too expensive for regular use by people with low income. Paratransit is an option for those who have a disability, but it is not an ideal solution as paratransit service has limited coverage and requires reservation long in advance of a trip.

In this dissertation, I address three major problems associated with public transit. First, improving the accessibility of complex public transit hubs for visually impaired travellers by sensing their location in the transit hub and guiding them to reach their desired bus or train. Second, I address the safety of the travellers – challenged by the outbreak of COVID-19 pandemic – inside a bus –

which is one of the most widely used public transit vehicles – by measuring their interpersonal distance for social distancing application. Third, improving the convenience of trip planing and safety of ticketing by localization of the passengers inside the bus for crowdedness monitoring and touchless ticketing. In this introductory chapter, I briefly touch on these problems and the approaches that I have taken to address them.

1.1 RouteMe2 and self-localization

Traveling by public transit, may be difficult for some people, and especially for those who need it the most. In many cases, the problem is one of information access. Managing a trip, especially one that requires one or more transfers, requires prior knowledge of where and when to catch each vehicle. While en route, travelers must acquire and process different types of information, such as which platform to stand on while waiting for a train, whether the bus vehicle that just arrived is the correct one. Travelers must maintain continuous awareness of where they are in the scheduled itinerary; make timely decisions (e.g. when to exit a bus); and devise contingency plans when something goes wrong (such as a missed transfer or a delayed arrival). Much of the information available to a traveler is in visual form, such as signs or displays, and thus inaccessible to those who are visually impaired (VI).

We are developing a system, called RouteMe2, that assists travelers with visual impairments to route between any source and destination points in complex transit hubs. Realization of this system improves the information accessibility required for a successful routing in these environment for those travelers.

While a sighted traveler may be able to spot the correct bus slot by looking at the posted signs, a blind traveler would likely need more directions to navigate the platform and find the place where to catch the bus. Indeed, for visually impaired travelers, who cannot rely on visual landmarks, routes need to be defined at a much finer scale than for sighted people. This is particularly the case in the open, when there are no readily available features that can be perceived by touch (such as a wall, which can be tracked using a long cane) and that can be used to follow a route. For the same reason, it is critical that users be spatially localized within a microroute, that is, pedestrian routes at a small spatial scale, with enough accuracy for the system to produce meaningful directions. I refer to this type of user localization as self-localization.

The accuracy of the self-localization in the outdoor environments is challenged by poor GPS reception due to tall nearby buildings that may obscure view of one or more satellites — a.k.a shading. Shading is very common in urban environments, and is a major cause of GPS failure. In order to mitigate the effect of shading, I statistically fuse the signals received from GPS as well as a small number of Bluetooth Low Energy (BLE) beacons installed in San Jose Diridon station. I further pair the statistical fusion with a Bayes discrete filter tracker to increase the self-localization accuracy.

Another challenge is clear and precise communication between the localization system and a visually impaired user who does not have access to the contextual spatiotemporal information available solely by the gift of sight. Contextual refers to the visual information that is available in the surrounding of the user within a certain area.

To satisfy the preceding demand, I leverage the insight I gained from our focus group study [3] and researching on the existing solutions. We found two core categories of issues faced by blind travelers: (1) spatial/location awareness, and (2) temporal/ time awareness. Configurability and accessibility were the most desired features requested for a new transit information system. Then, I follow a user experience (UX) design process. It is an iterative process that involves sketching and generating storyboards, wire-framing and prototyping, and developing and implementing the designs.

1.2 Interpersonal proximity detection for social distancing

During my study the COVID-19 pandemic has affected virtually all enterprises in the private and public sector. High risk of contagion put the safety of public transits in question for everyone. It soon became obvious that there must be some regulatory mechanisms to increase the safety of the public areas. One of these regulatory mechanisms was social distancing to reduce the spread of the air-born virus. The importance of social distancing in the safety of the travellers provoked my motivation to study interpersonal proximity inside a public transit vehicle. In particular, I address the specific problem of detecting the presence of another individual within distance thresholds of 1 meter and of 2 meters, since these are interpersonal distances usually considered when establishing the risk of contagion [4,5]. I consider the case in which two individuals are standing or sitting at certain distance from each other for a period of time. This is representative of typical contagion scenarios, such as sitting in nearby seats in a bus vehicle. I present an experimental comparative analysis of mechanisms that use measurement disparity of the received signal (RSSI) from fixed BLE beacons installed in a vehicle and viz-a-viz the direct measurement of RSSI from another nearby smartphone. There are interesting challenges in this problem that makes it more attractive. Buses are subject to significant signal reflections due to their metal walls and mobility and presence of the crowd inside the bus.

1.3 Effortless ticketing and crowdedness monitoring

Effortless (or implicit [6]) ticketing, refers to methods that enable payment of the correct fare as triggered by the mere presence of the user inside the vehicle. Current touchless fare payment technology still requires travelers to approach a near-field communication (NFC) reader or possibly a QR reader [7] located in the vehicle. This creates "accumulation points" of social proximity, which may slow the flow of passengers entering the vehicle and thus increase boarding times. A real effortless ticketing system would not require the users to take any actions, except for starting an app in their smartphone. It would automatically identify the vehicle boarded by the passenger (Be In/Be Out, or BIBO, modality [6]), and charge the correct fare. Upon boarding the bus vehicle or train car, the user would receive a notification (e.g., via a vibration) from the system that the vehicle has been identified, and that that ticketing is taken care of.

The same mechanism that enables effortless ticketing can be used to assessing and track the distribution of passengers in a vehicle. Crowdedness monitoring has received increased recent attention [8]. I envision a system that measures not only the approximate number of passengers, but also their spatial distribution in the vehicle. This could be very useful when deciding which door to enter a vehicle from. For example, if riders in a train cart or bus vehicle are concentrated in the front half, a passenger waiting at the stop may decide to enter from the back door (see Fig. 1.1). This information could also be very valuable for transit agencies, which can put in place provisions to ensure a uniform distribution of passengers

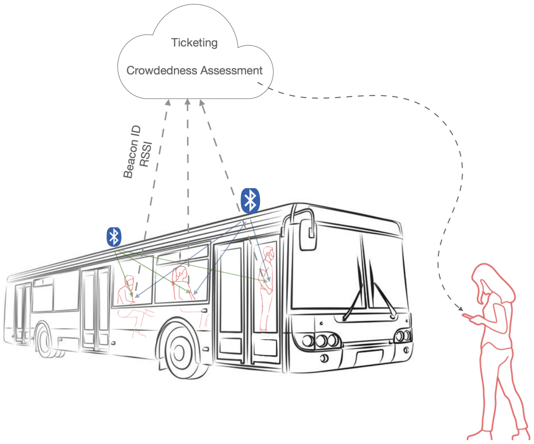


Figure 1.1: The envisioned system uses RSSI data recorded from the in-vehicle BLE beacons for passenger positioning, enabling effortless fare payment and crowdedness assessment.

in their vehicles.

To that end, I use BLE beacons as the underlying technology for both services considered (effortless ticketing and crowdedness monitoring). BLE beacons are inexpensive and unobtrusive. Battery operated models (e.g., Kontakt Tough Beacon TB18-2) can last up to 80 months on a battery charge, and require no vehicle retrofitting (including wiring) nor maintenance during this period. I present results from a preliminary study on passenger localization within a bus vehicle using BLE beacons. I instrumented a campus shuttle bus with four BLE beacons, and conducted multiple data collection session. RSSI data was collected from all beacons while the experimenter sat in different location within the vehicle.

The thesis is organized as follows: in chapter 2.4 I review the most current studies on indoor and if available outdoor localization. In section 2.5, I present an overview of RouteMe2 app and its architecture. Then I segue to the deterministic models I experimented with that seemed promising for solving the self localization problem. You will see that these model would not be suitable for a complex outdoor environment such as VTA Diridon Station's transit hub. Then, I explain the stochastic model that I used to address the challenging problem of self localization in that environment. Section 2.10 presents the data collections, results, discussions, and my observations. I conclude this work in section 2.13.

In chapter 3 section 3.3 I briefly review other accessible navigation systems built for people with visual impairments. In section 3.4 I present the user interface design thought process. Features and functionalities of the app is presented in section 3.5. Finally, I conclude the chapter in section 3.6.

Chapter 4 has two sections. Interpersonal proximity detection is presented in section 4.2. In section 4.2.1 I provide the related work on interpersonal distance detection studies. Techniques and experiments are presented in sections 4.2.2 and 4.2.3. In section 4.2.4 I provide discussion and conclusions on this work. In section two of this chapter 4.3 effortless ticketing and crowdedness monitoring is described. In section 4.3.1 related work is presented. In section 4.3.2 I walk you through the experiments we run on our collected data in a shuttle bus. The discussion and conclusions is then presented in section 4.3.3

Chapter 2

Localization Using BLE in Diridon Station

2.1 Introduction

I am developing a system, called RouteMe2, that assists travelers with visual impairments to route between any source and destination points in complex transit hubs. Realization of this system improves the information accessibility required for a successful routing in these environment for those travelers.

RouteMe2 is embodied in a smartphone app and a cloud server system. The software in the app and in the remote server work in tandem to track the traveler, and to provide trip-specific and location-aware information, in a format that is convenient for the user. For example, blind users may receive directional guidance in the form of synthetic speech, with the app guiding them to specific places (e.g. the location of a bus stop) while leveraging landmarks that may be perceivable without sight (such as a bench that is known to be located at the bus stop). For people with cognitive impairment but with usable sight, the app may provide simple directions at each step of the way, possibly relying on visible landmarks (“stand next to the red pole with a bus sign on top"), and allowing them to read or hear the directions as many times as desired. RouteMe2 uses existing trip planning APIs (such as Google Directions or OpenTripPlanner) to determine a route, and tracks the user through the route, re-routing when necessary.

Unlike existing routing and tracking apps already available (e.g. Google Maps, Apple Maps, Transit, Moovit, Citymapper), RouteMe2 is able to generate *micro-routes*, that is, pedestrian routes at a small spatial scale. A typical pedestrian route (e.g. from Google Map) normally specifies paths on roads or pedestrian routes to a transit station or bus stop. Sometimes, when this information is available (from a GTFS or NeTEx file), these routes can even specify the bus slot in a large transit center. But rarely, if ever, do these generated routes include small-scale detailed spatial information. For example, Fig. 2.1 shows a pedestrian route to a bus stop. The route ends at the edge of the multi-slot platform. While a sighted traveler may be able to spot the correct bus slot by looking at the posted signs, a blind traveler would likely need more directions to navigate the platform and find the place where to catch the bus. Indeed, for blind travelers, who cannot rely on visual landmarks, routes need to be defined at a much finer scale than for sighted people. This is particularly the case in the open, when there are no readily available features that can be perceived by touch (such as a wall, which can be tracked using a long cane) and that can be used to follow a route. For the same reason, it is critical that users be spatially localized within a microroute with enough accuracy for the system to produce meaningful directions.

RouteMe2 comprises different components and features including user registration, route definition, self localization within microroutes, tracking, local routing, and rerouting when needed. Disabled users or their caregivers can register and

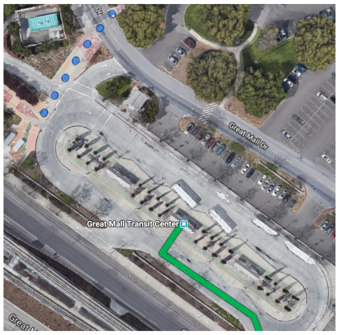


Figure 2.1: Detail from a route generated by Google Map. A pedestrian route, shown by blue dots, leads to a bus hub, from which the desired bus line departs (green line). Finding the route from the edge of the bus hub to the desired bus slot may be challenging for a blind traveler.

create an account in RouteMe2 system. They define their routes in the system and they chose their desired route to start. The app uses Google direction API for routing (i.e. itinerary steps). When the user arrives at the station the app switches to the local microrouting mode. In this mode, the user is guided through a sequence of Transit Locations (TL). Each TL is a polygon − represented by connected geographic points − that specifies a small subspace of the walkable area in the station. The user receives location updates within each TL as he/she walks along the route. The self-localization component in the server is responsible for accurately estimating the current location of the user. In this chapter self-localization component takes advantage of the signals received from a number of beacons installed in the complex transit hub (*Valley Transition Authority* VTA Diridon station in our case) fused with the GPS estimated location. All of these signals are received by the phone and are sent to self-localization component in the server. In response, the location of the user in a microroute along with detailed spatial information such as nearby landmarks, kiosks, benches, turns , etc. are sent back to the phone. The estimated current location is visually shown in a map and is also provided by means of synthetic speech to guide the users with visual or cognitive impairments.

2.2 Contribution

To fully realize the RouteMe2 system I addressed many different challenges and problems as described herein.

I address the case of an outdoor transit station with poor GPS reception due to tall nearby buildings. As well known, the presence of tall structures may obscure the view of one or more satellites (*shading*, also known as urban canyon effect). If fewer than 4 satellites are visible, signal from the remaining satellites can only be received via multipath, generating localization errors. Shading is very common in urban environments, and is a major cause of GPS failure. My preliminary experiments showed $10m$ average distance error if the travelers are to use GPS for their navigation in the station and the surrounding area. That makes the GPS navigation inapplicable for even sighted travelers, set the visually impaired people alone. In order to mitigate the effect of shading, I propose the use of Bluetooth Low Energy (BLE) beacons. BLE beacons are a popular tool for localization in indoor, GPS-denied environments. The use of BLE beacons in the outdoors, for situations with poor GPS reception, has received much less attention by the research community.

Transmission models are popular for distance estimation. I studied estimation of the distance between beacon and phone to understand the accuracy of range-based distance estimator model (transmission model). Despite popularity of transmission model, my experiments with this model in different locations, weather conditions, angles, heights, beacon orientations proved this model unsuitable for microrouting visually impaired people. See section 2.8.1 for more detail.

Power regression models are another paradigm that has been considered for indoor self localization. I studied estimation of the distance between beacon and phone to understand the accuracy of power regression distance estimator model. I ran the same experiments I did with transmission model to estimate the distance with the phone in different locations, weather conditions, angles, heights, beacon orientations. As suggested by my result this model also did not seem to be a reliable distance estimator. Please see section 2.11 for more detail.

Although, beacons are promising sensors to leverage for realizing the micro-routing prospect, they deemed challenging due to various types of noises including weather conditions, direction of the receiver, obstruction of the receiver as the user carries the receiver device (i.e. mobile phone), and presence of the crowd i.a. I explore the combination of spatial information from GPS and BLE beacons via statistical fusion. I show that, with a proper modeling of the error distribution of the signals involved (spatial location from GPS, RSSI from beacons), it is possible to achieve substantially lower localization error than when using either modality in isolation.

My studies on a single beacon model revealed that signal broadcasting pattern of beacons is subject to change based on the spatial orientation of the beacon. Peak *et al.* [9] had the same observation in indoor environment and studies this behavior on beacons and phones from various vendors.

Another important source of variation is the fact that the person can hold the phone in different positions including in-hand and in-pocket. Also, the beacon could be obscured by the persons body depending whether they are facing the beacon or not as they hold the phone while using the tracking app. To consider these situations I collected all the mentioned situations and combined in both transmission and power regression models training and testing phases.

VTA Diridon station is a complex transit hub with $1750m^2$ area. A challenge is to collect enough data for building a model for microrouting due to the size of the deployment site and presence of travelers in the station. This mandates the model to be as sample efficient as possible. I collected the time-stamped data for training and testing the model by walking at an approximately constant velocity in traversable areas and extracted the locations by knowing the starting point, end point, and the velocity. The advantage of this type of data collection is its time efficiency compared to stationary data collection by standing in different points along the traversable areas in a large environment.

Existence of unsafe areas in the station such as tracks on the railways mandates a thoughtful routing calculation to enable a visually impaired user for safely navigating the area. I defined traversable areas to be the area where the visually impaired person can safely navigate through. The self-localization algorithm estimates current location of the user with respect to traversable areas to deterministically avoid guiding the person to unsafe areas. Whereas map applications such as Google Map or Apple Map simply do not take this scenario into their consideration.

I observed sporadic jumps in the estimated location due to multimodality of the posterior probability of location based on BLE beacons and GPS data. To that end, I implemented a spatial tracker based on discrete Bayes filtering. The tracker updates a posterior probability distribution of the user's location over a grid of traversable (walkable) locations. I used this tracker to circumvent the problem of multimodal probability distribution of the location state solely estimated by means of maximum log likelihood.

2.3 Chapter Organization

In chapter 2.4 I review the most current studies on indoor and if available outdoor localization. In section 2.5, I present an overview of RouteMe2 app and its architecture. Then I segue to the deterministic models I experimented with that seemed promising for solving the self localization problem. You will see that these model would not be suitable for a complex outdoor environment such as VTA Diridon Station's transit hub. Then, I explain the stochastic model that I used to address the challenging problem of self localization in that environment. Section 2.10 presents the data collections, results, discussions, and my observations. I conclude my work in section 2.13.

2.4 Background and Related Works

Riding public transit can be confusing for everyone, especially in an unfamiliar environment. One needs to figure out which transportation lines to take to reach a destination, when and where to catch a bus or a train, when to exit, and how to negotiate transfers. For those with sensorial or cognitive disabilities, these problems become even more daunting. Several technological approaches have been proposed to facilitate the use of public transit for people with visual and cognitive impairments.

Previous studies have shown that people with visual impairment experience difficulties at determining the route and schedule information, purchasing fare, finding the correct bus-stop location, getting on the correct bus, and getting off at the right stop [10–14] focused on identifying the correct bus to board when waiting at a bus stop, while the systems described in [15–17] provided alerts for an upcoming stop while riding the bus. Hara *et al.* [18] proposed gathering spatial and temporal information from different patterns of mobility and travel time using smart card and GSM data. They aimed at building a public transportation system that could adapt to different travel patterns for different situations. Tribby *et al.* [19] proposed a high resolution spatio-temporal, Geographic Information System (GIS) based public transit network model to measure different models of travel time, such as waiting time at bus stop and transfer times between routes. A variety of solutions have been proposed to help people with blindness and with limited vision, including providing non-visual information about the location of bus stops. For example, Azenkot *et al.* [12] developed GoBraille, a system that uses crowdsourcing to gather detailed information about the location of stops (a similar system is StopInfo [20]). This prior work shows that there is a need for people with limited or no vision to be constantly aware of where they are in reference to their travel goals, as well as to obtain the information that is necessary to utilize public transit effectively. However, these prior studies do not offer in-depth knowledge and detail to the level that is necessary to make correct design decisions on the best tool for accessible public transit. Motivated by this observation, a focus group study was conducted to observe group dynamics of several participants with blindness and visual impairments [3]. Location and time awareness were the two core themes emerged from the grounded theory analysis of the issues faced by blind travelers. Location awareness deals with being aware of one's geographical position in reference to the public transport throughout the entire trip. Most participants reported situations with loss of location awareness due to multiple reasons, such as knowing whether or not they are in the right vehicle, whether they are waiting for a bus at the right stop, whether the bus vehicle they are waiting for is close or far, and whether they stand next to the entrance door of the bus vehicle or train car. Some of the main themes that emerged during the focus group study proceed as follows. Some participants complained that routes (including the list of stops) and schedules are not clearly communicated. Finding the exact location of bus stops and train platforms was one of the main challenges for the five participants who were completely blind. This includes understanding which side of the street the bus stop is located at, and whether one needs to cross the street to reach it. Finding the correct train platform is also challenging. In addition, knowledge of the layout of a stop is important when one needs to negotiate a transfer. Participants mentioned that in these situations they often rely on sighted travelers, when available. Locating doors of buses or trains with multiple units was mentioned as a challenging task, especially for the local subway system (BART). Maintaining awareness of one's surroundings is particularly important. Participants shared experiences of walking in the wrong direction after leaving a train or a bus, as they had no clear idea of the surrounding area. Catching the right bus or train and knowing they are in the right one was an issue mentioned multiple times in the discussion. Excessive ambient noise, and wrong or incomplete announcement from the vehicle's speakers, may cause loss of state awareness in these situations.

As mentioned earlier in the introduction chapter, in order to provide such detailed spatial information microrouting is necessary for people with visual and cognitive impairments. A visually impaired person would have detailed location awareness if we had enough knowledge about his/her surrounding. This may not be accomplished without knowing the location of the person. It is critical that users be spatially localized within a microroute with enough accuracy for the system to produce meaningful directions.

Self localization problem constitutes a major problem in solving the blind travelers issue. With the growth of smart cities and Internet of Things (IoT) applications, development of accurate, scalable, and reliable outdoor localization systems is one of the recent years' challenges for many scientists and engineers [21]. Global Position System (GPS) is widely used for outdoor positioning systems [22, 23]. However, GPS is prone to error due to different factors including signal blockage, atmospheric conditions, and multi-path interference which occurs in the presence of tall buildings and mountains.

There is increasing interest in systems that enable self-localization in GPS-denied environments (e.g. indoors). While in principle it could be possible to use power decay models [24,25] to estimate the distance to a beacon from the measured RSSI from that beacon, then self-localize via multilateration, in practice this is extremely challenging [9] due to issues such as multipath fading (an effect of signal reflection from nearby surfaces) and variations in time of the signal power. For this reason, it is customary to instead "learn" a mapping from the set of received RSSI signal from one or more beacons, to the user's location. This mapping is learned from measurements taken at multiple, known locations, a process called *fingerprinting* [26–29].

Localization using RSSI value from Wi-Fi access points (AP) has been widely studied [26–28, 30, 31]. This approach leverages the widespread availability of Wi-Fi APs in public environments. However, one generally has no control on the actual density of placements of APs (meaning that some areas of interest may not be covered), or on other factors such as APs being disconnected or moved after fingerprinting. BLE beacons represent a popular alternative to Wi-Fi APs [32–36]. BLE beacons are generally inexpensive, and being battery-operated they can be placed where desired without wiring concerns.

Parameswaran *et al.* [37] ran different experiments with the aim to prove or disprove that RSSI can be used as an indicator of distance between motes in a sensor network. They collected 30 measurements at different distances/angles between receiver and sender. One of their findings was lack of reliability when tested in different directions. Another finding was error in measured RSSI value increases when distance increases. They state that Gaussianity assumption does not necessarily hold. Despite these challenges that RSSI approach encounters, researchers improved the accuracy of localization by proposing new algorithms or/and combination of different approaches in a multi-sensor environment to design more robust systems [38]. Next, I explored the studies used BLE beacons and their suggestions of deployment settings.

Faragher *et al.* [32] compared BLE and WiFi for indoor positioning. They postulate that fast-fading effect is more significant in BLE sensors compared to WiFi. Kriz *et al.* [39] showed that the higher advertisement frequency and denser beacon deployments yield higher positioning accuracy. Budina *et al.* [35] proposed a method of iBeacon optimal distribution for indoor localization. Their study suggested that full coverage of the space and detecting enough devices with enough signal intensity improve the localization accuracy. Castillo-Cara *et al.* [36] studied some of the beacon parameters, such as transmission power, density, and topology. They made the following recommendations: setting transmission powers to low or medium, avoiding positions near windows, enough gaps between beacons, and distance of at least $6m$ between beacons, being aware that the material composing the walls affect the signal propagation. Rezazadeh *et al.* [40] showed that vertical and horizontal positions of beacons affect their broadcasting range of the beacons. By alternating the positioning of the beacons they were able to increase the overall coverage in the region of interest.

Here, I mention couple of the studies and systems that use BLE beacons and WiFi sensors for indoor/outdoor localization. NavCog3 [41, 42] is a turn by turn navigation system for people with visual impairments for indoor environment. They use inertial sensors (accelerometers and gyroscopes) on a mobile device and RSSIs from Bluetooth Low Energy (BLE) beacons for localization. They use two models for localization and particle filter for tracking. A motion model uses inertial sensors to detect a location and observation model that computes the likelihood of the sensor measurements given the location or state. They proposed Localization Integrity Monitoring (LIM) module to efficiently evaluate the estimated localization by detecting unreliable state and restarting the system to initialization state. The mean localization error was between 3.0m to 1.6m when tested on 5 floor shopping mall. Zafari *et al.* [34] used cascaded Kalman Filter – Particle Filter (KFPF) algorithm for indoor localization using iBeacons. Their model outperformed their previous work [43] of proximity detection using only particle filter by 28.16% and 25.59% in 2D and 3D environment respectively. Use of particle filter comes with the cost of higher computational complexity. For that reason they proposed two Server-side Running Average (SRA) and Server-side Kalman Filter (SKF) algorithms. They improved proximity detection of iBeacons by 29% and 32% compared to Apple's Core Location framework. Jeba *et al.* [44] developed a monitoring system for elderly people inside a smart home environment. They proposed 2-dimensional tracking system. They use RSSI-based localization to estimate the location of movable objects in an equipped space with no obstacles. Then, movement of the receiver node with different speed is judged with a Markov model. They deployed two sensor nodes in two fixed locations $(0, 0)m$ and $(0, 12)m$ in $12 \times 12m$ Region of Interest (ROI) which is classified into $(3 \times 3)m$ shells. A receiver sensor is installed on a small robot that can move randomly with constant speeds.

Ran *et al.* [45] built navigation system for indoor/outdoor localization. It is designed for people with visual impairments. Their outdoor localization uses DGPS. For indoor positioning they use ultrasound positioning service. They communicate with visually impaired person via voice commands generated based on spatial information obtained from their testing site. User need to carry the wireless sensors during the testing.

My work uses BLE beacons for localization in an environment in which GPS data is unreliably available. I am not aware of prior work that leverages both sources of information (GPS and BLE beacons) to improve self-localization in these scenarios, even though similar situations are relatively frequent in urban environments. The self-localization algorithm estimates and track the user's location with respect to safe walkable areas to deterministically avoid guiding the person to unsafe areas. RouteMe2 is able to generate microroutes and provides detailed spatial information in complex transit hubs by microrouting. The proposed model can be deployed on commercial personal mobile phones and in any complex transit hub with such dynamicity and noise.

2.5 Model

In this chapter, I first walk you through the overview of RouteMe2 system in section 2.6. Detail of the GPS and beacon sensors that I used to predict the location of the phone is provided in section 2.7. My first attempt was to understand whether I can use the deterministic RSSI-based models of beacon to estimate the location. These models are described in section 2.8. I show that they are not quite reliable estimators of location in outdoor environments as demonstrated in results chapter (ch. 2.10). Finally, I introduce the stochastic model that I used for location estimation in section 2.9 by providing all the background and mathematical derivation of the model along with my rationale behind why it is a good model for self localization.

2.6 RouteMe2 Architecture Overview

RouteMe2 comprises different components including user registration, route definition, self localization within microroutes, tracking, local routing, and rerouting when needed. Figure 2.3 shows the overview of the current version of RouteMe2 app. User logs in into RouteMe2 app using the required credentials (fig. 2.3, a). RouteMe2 app sends the credentials to the server and retrieves predefined trips from database and send them to the app. Next, user can select a trip (fig. 2.3, b), then app sends the trip information to the server and the server communicates with Google Direction API to get the itinerary. Server sends the itinerary to the app. App shows the route information on the map (fig. 2.3, c). When the user arrives at the station, the app switches to local microrouting mode (fig. 2.3, e). As mentioned earlier, microroute is a pedestrian route at a small spatial scale. A typical pedestrian route may specify paths on roads or pedestrian routes to a transit station, bus stop, or bus slot in a large transit center.

A traversable area is a safe walkable area in a continuous location space that is segmentized with transit locations. A transit location (TL) is a polygon – represented by geographic points – that specifies a small subspace of traversable area. Figure 2.2 shows an example of microroute, TL, and traversable area in the iPhone app.

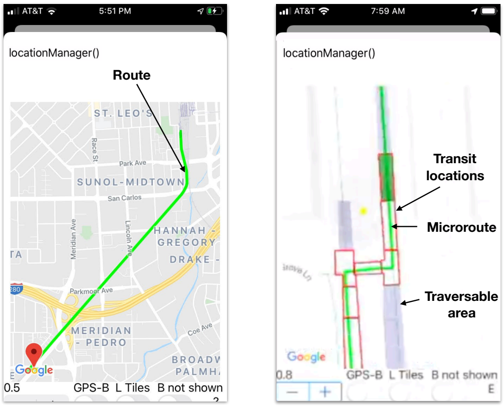


Figure 2.2: Route, transit location, microroute, and traversable area are annotated. Red polygons represent TLs. Each TL is a segment of traversable area – safe walkable area. Green line represents a microroute. A microroute is a pedestrian route at a small spatial scale.

Throughout microrouting, user receives location updates as he/she walks along the route. Figure 2.4 shows the self-localization component of the system. The self-localization component in the server is responsible for accurately estimating the current location of the user. This component takes advantage of the signals received from a number of beacons installed in the complex transit hub – *Valley Transition Authority* (VTA) Diridon station in this study – fused with the GPS estimated location. All of these signals are received by the phone and are sent to self-localization component in the server. In response, the location of the user in a microroute along with detailed spatial information such as nearby landmarks, kiosks, benches, turns , etc. are sent back to the phone. The estimated current location is visually shown in a map and is also provided by means of synthetic speech to guide the users with visual or cognitive impairments.

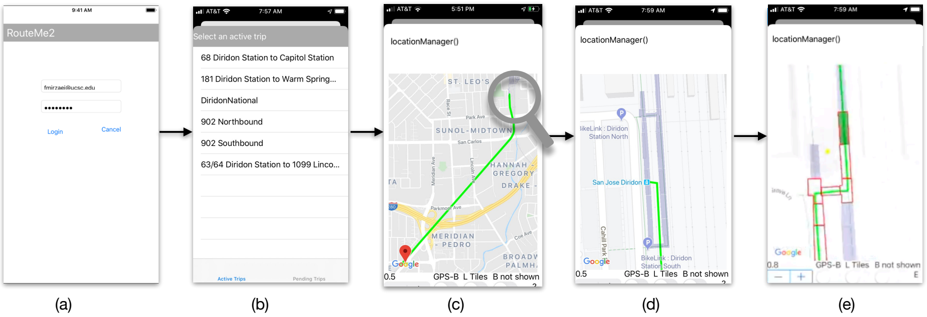


Figure 2.3: User interface of the RouteMe2 app is demonstrated. a) User logs in into RouteMe2 app using the required credentials. RouteMe2 app sends the credentials to the server and retrieves predefined trips from database and send them to the app. b) Next, users can select a trip, then app sends the trip information to the server and the server communicates with Google Direction API to get the itinerary. Server sends the itinerary to the app. c) App shows the route information in the form of map. d) Source (starting point) of the itinerary route is shown by zooming into the source of the itinerary route. The source shown here is what is returned by Google Direction API and is not the exact source. e) When the user arrives at the station, the app switches to the local microrouting mode. In this mode, the user is guided through a sequence of microroutes to catch the train or bus at the exact desired location.

Figure 2.4 shows the architectural overview of the current version of RouteMe2 app and flow of information throughout the routing. The self-localization component resides in the remote server in IBM cloud. User's phone receives GPS and beacons' data and sends them to the server. The server computes the estimated location. Then, it checks the estimated location against the current route. If estimated location was not on route, the route gets updated. The server returns the estimated location along with microrouting information to the user's phone. Location and route are displayed on the app's map.

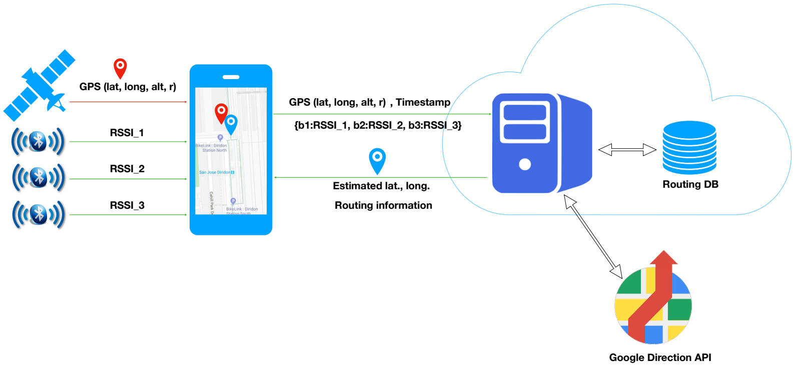


Figure 2.4: Architectural overview of the current version of RouteMe2 app and flow of information throughout the routing. User's phone receives GPS and beacons' data and sends them to the server. The server computes the estimated location. Then, it checks the estimated location against the current route. If estimated location was not on route, the route gets updated. The server returns the estimated location along with microrouting information to the user's phone. Location and route are displayed on the app's map.

2.7 Sensors Considered

I considered using GPS and BLE beacons as the source of information required for self localization. Sections 2.7.1 and 2.7.2 present a background about GPS and BLE beacon and the rationale behind using them for estimating the location. describes how the Geographic Coordinate Systems is projected onto a Cartesian coordinate in order to show the location on the phone's map.

2.7.1 Global Positioning System (GPS)

GPS is a satellite navigation system developed and maintained by the United States Department of Defense. It includes 24 satellites in semi-geosynchronous orbit providing continuous global coverage [46].

GPS receivers can generally be classified into one of three categories: survey, mapping, and consumer grade. Survey-grade GPS is capable of determining locations to within 1cm of true position [47], but requires operator expertise and a substantial operating budget because instrument costs can exceed $\$25,000$. Survey-grade GPS also requires satellite signal reception that is often unattainable in the receiver site. Mapping-grade GPS receivers can return accuracies typically within 2˘5m of true position, depending on the quality of the equipment and operator skill, with instrument costs ranging from $\$2,000$ to $\$12,000$. Although mapping-grade GPS can be somewhat more forgiving than survey grade in terms of acceptable satellite reception and required operator skill, the price of these units is still prohibitive to many potential users. In contrast, consumer-grade GPS receivers are now available at a very low price and ubiquitous in consumer mobile phones. Consumer-grade GPS manufacturers commonly assert that measurement accuracies of this equipment should be within 15˘20m of true position [48]. Wing *et al.*tested the accuracy and reliability of consumer-grade GPS receivers in a variety of landscape settings. Among the top GPS performers, they determined that users could expect positional accuracies within approximately 5m of true position in open sky settings, 7m in young forest conditions, and 10m under closed canopies [48]. Consumer-grade GPS receivers that are available in phones typically provides latitude, longitude, altitude, and radius of accuracy. Latitude, longitude, and altitude are described in the next section (2.7.1). Section 2.9.3 describes the detail of how the radius of accuracy is used in this study.

From Geographic Coordinate to Cartesian

On geographic coordinate system, every location on earth can be specified by latitude, longitude and altitude. Map-makers choose a reference ellipsoid with a given origin and orientation to disambiguate the direction of *vertical* and the *horizontal* surface according to their need for the area to be mapped. Geodetic datum (a.k.a. terrestrial reference system) is the mapping between the spherical geographic coordinate system onto that reference ellipsoid. Datums may be global, meaning that they represent the whole Earth, or they may be local, meaning that they represent an ellipsoid best-fit to only a portion of the Earth. Although datums are subject to errors over time due to continental plate motion, subsidence, and diurnal Earth tidal movement caused by the Moon and the Sun. These changes are insignificant if a local datum is used, but are statistically significant if a global datum is used [49]. Different datums have been defined including World Geodetic System (WGS84, a.k.a. EPSG:4326) which is the default datum used for the GPS and is used in iPhone devices which is of my concern in this study.

The latitude and longitude provided on iPhone is defined on WGS84 datum and I refer to it as GPS coordinate system. The reader shall notice that this is not equivalent to the real spherical geographic coordinate of the Earth. Instead, it is a geographic coordinate on WGS84 datum provided by GPS on the phone (iPhone in this study).

The points on GPS coordinate system are represented by latitude ϕ), longitude λ, and height(h). In order to transform GPS coordinates to the local Cartesian coordinate system with an origin, I employ East, North and Up (ENU) transformation. ENU is in fact a Cartesian coordinate system. ENU is the most common transformation in the tracking and targeting application. First, I transform geodetic coordinate system to Earth-Centered, Earth-Fixed (ECEF). Then, I transform ECEF to ENU coordinate system. ECEF, is a geographic and Cartesian coordinate system that represents positions as X, Y, and Z coordinates. The point (0, 0, 0) is defined as the center of mass of Earth. Figure 2.5 shows the coordinates of ECEF axis and ENU axis [1].

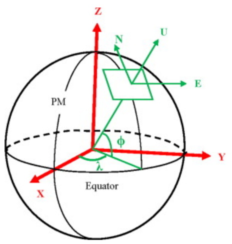


Figure 2.5: WGS84, ECEF, and ENU coordinate systems for the Earth and their transformation relationships (PM line is the Prime Meridian; and are latitude and longitude in WGS84; X,Y,Z for ECEF; and E,N,U for ENU [1].

Geodetic coordinates (ϕ, λ,) can be converted into ECEF coordinates using the following equation:

$$\frac{p}{\cos(\phi)} - \frac{Z}{\sin(\phi)} - e^2 N(\phi) = 0 \tag{2.1}$$

where $p = \sqrt{X^2 + Y^2}$, $e^2 = 1 - \dfrac{b^2}{a^2}$,

$$N(\phi) = \frac{a}{\sqrt{1 - e^2 \sin^2(\phi)}} \tag{2.2}$$

where a, b are equatorial radius, polar radius respectively. WGS-84 datum uses $a = 6378137.0$ m and $b = 6356752.314245$ m values.

ENU is one of the Local tangent plane coordinates (LTP) variation. LTP, is the coordinate system that consists of three coordinates. A local plane is tangent to the earth's surface where its origin located at local origin I define. East coordinate is along the local eastern axis, North axis is along the northern axis, and Up represents the vertical position. Vertical position can be up or down.

The relation between ECEF and ENU and rotation matrix is defined as below.

$$\mathbf{p}_{ENU} = R(\mathbf{p}_{ECEF} - \mathbf{p}_{Ref}) \tag{2.3}$$

Where

$$\mathbf{R} = \begin{bmatrix} -\sin(\lambda) & \cos(\lambda) & 0 \\ -\sin(\phi)\cos(\lambda) & -\sin(\phi)\sin(\lambda) & \cos(\phi) \\ \cos(\phi)\cos(\lambda) & \cos(\phi)\sin(\lambda) & \sin(\phi) \end{bmatrix}$$

and $\mathbf{p}_{ECEF}$ is the coordinates of ECEF of a point in WGS84 coordinate system which can be obtained from eq. (2.1). $\mathbf{p}_{ref}$ is the local reference (origin) point in ECEF coordinate system.

2.7.2 Bluetooth Low Energy (BLE) Beacon

GPS is widely used for outdoor positioning systems [22, 23]. However, GPS itself has error due to different factors including signal blockage, atmospheric conditions, and multi-path interference which occurs in the presence of tall buildings and mountains. I investigate whether fusing GPS data with other sensors would improve localization. Range-based are the most common approaches for self-localization. The range-based schemes are defined by the protocols that use distance estimates for location computation. Recently, Bluetooth Low Energy (BLE) protocol sensors grabbed scholars' attention for development of low cost, energy efficient, precise and accurate indoor positioning system. Ranged-based location estimation approaches are further explained in the following sections.

BLE beacons use radio propagation model that use Receive Strength Signal (RSS) and the known location of beacons to estimate the location of sensor node which is a phone in this study.

2.8 Deterministic RSSI Models Investigated

I first investigated two important deterministic RSSI models – namely transmission model and power regression model – proposed in literature that are provided in this section. The results of my investigation are presented in section 2.11

2.8.1 Transmission Model

The goal of this model is to estimate the distance between a beacon and a receiver. A receiver can be incorporated in a phone. A phone at position (x, y) receives broadcasted RSSI from a beacon at (x_i, y_i) position and estimates its location using the received power (RSSI). Transmission model is a widely accepted model in the literature [50]. Despite that, as I will show in section section 2.11.4, it is not a reliable model for distance estimation in Diridon outdoor environment. The estimation error increases when the distance increases. Also, my observation on the distribution of RSSIs in various distances suggests that due to the presence of noise or inner beacon signal interference one path-loss coefficient (described here) is not enough for reliably estimating the distance.

Transmission model is also called path loss model. In this model, the received power p_i (dBm) is explained as in eq. (2.4).

$$p_i = p_0 - 10n \log(r_i/r_0) \tag{2.4}$$

where n is a path loss or transmission constant that varies based on transmission environment. Papamanthou *et al.* [50] claimed that for outdoor environment the value of n is between 2 and 4. r_0 is the reference distance and p_0 is the received power at reference distance. The power is measured by means of RSSI. p_0 can be obtained by taking the average of RSSI values collected at reference distance. For instance, 1 or 2 meters away from a beacon. Alternatively, p_0 can be obtained by factory's default received power which changes by altering the transmission power of the beacon. p_i is the ideal received power and r_i is the distance from the beacon. I replace p_i with $\hat{p_i}$ and r_i with $\hat{r_i}$ in eq. (2.4) which yields eq. (2.5).

$$\hat{p_i} = p_0 - 10n\log(\hat{r_i}/r_0) \tag{2.5}$$

$\hat{p_i}$ is estimated received power that accounts for channel fading due to the noise imposed in a real world environment. Fading in this wireless communication channel is mostly referred to a random process [50] represented by random variable x in

$$\hat{p_i} = p_i + x \tag{2.6}$$

where random variable x represents the medium-scale channel fading and is typically modelled as Gaussian zero-mean with variance σ^2 (in dBm) [50]. x is independent from distance. $\hat{r_i}$ is the estimated distance I describe later in eq (2.8). I can derive the relationship between estimated distance $\hat{r_i}$ and real distance r_i by combining the equations (2.4), (2.5), (2.6) and get

$$\hat{r_i} = r_i^{10^{x/10n}} \tag{2.7}$$

By using (2.7) and inverting (2.5) I can compute estimated distance as

$$\hat{r_i} = 10^{\frac{p_0 - \hat{p_i}}{10n}}.r_0 \tag{2.8}$$

I can calculate the relative error $\mathcal{E}_i$ by eq. (2.9), where $\hat{r_i}$ is the measured distance, and r_i is actual distance.

$$\mathcal{E}_i = \frac{\hat{r_i}}{r_i} - 1 \in [1, +\infty). \tag{2.9}$$

by inserting $\hat{r_i}$ from eq. (2.7) into eq. (2.9) I also get

$$\mathcal{E}_i = 10^{x/10n} - 1. \tag{2.10}$$

2.8.2 Power Regression Model

A distance calculation equation proposed by AltBeacon Android library [51, 52] has the advantage of considering every factor that influence the radio wave propagation on a given context (i.e. environment) [53]. I call this model power regression model hereafter.

$$\hat{r_i} = A.(\frac{\hat{p_i}}{p_0})^B + C \tag{2.11}$$

where $\hat{r_i}$ is estimated distance, $\hat{p_i}$ is received power, p_0 is reference received power at reference distance. A, B, and C are environment coefficients that vary based on different medium. The coefficients need to be trained every time the medium and device changes. If the ratio of RSSI over reference RSSI is less than one, then the distance is the ratio to the power of 10, otherwise the equation 2.11 holds.

Path loss n varies in different conditions such as holding phone on hand or in pocket. Power regression model needs a lot of data from the environment to obtain the A, B, and C coefficients. This model is also not applicable in this study as the coefficients need to be trained every time the medium and device changes. Outdoor environment is dynamic and it changes constantly due to different factors including weather conditions, movement of buses and trains, people, etc. Hence, obtaining the coefficients is not feasible as a fixed coefficient would not fit for too many environmental conditions in the complex outdoor setting of Diridon station.

2.9 Proposing Stochastic Model

The objective of self-localization is to estimate the phone's location state from sensor measurements (i.e. beacon RSSIs and GPS data) as well as all past sensor measurements and location changes of the phone from one location to another (i.e. tracking).

As I explained in the previous two sections and backed by my experiments and results in section 2.11, these two models (i.e. transmission model and power regression model) cannot be used reliably for location estimation in the presence of various noises in a complex transit hub such as Diridon station. What is needed, is a model that can tolerate all kinds of environmental noises and uncertainties to reliably serve people with special needs.

For that reasons, a stochastic model turns out to be a more appropriate model to solve the self-localization problem in a complex transit hub. In this section, I fully describe this model in detail. The model needs to be precise enough to accommodate for local microrouting the visually impaired people and at the same time be computationally efficient. Discretizing the location state space in the station by defining a grid with granularity of $1m$ would meet my objective (see section 2.9.1 for details of the grid). In sections 2.9.2 and 2.9.3, I describe how the RSSI and GPS are modeled and, then, I explain the mechanics of fusing the two in section 2.9.4. As you will see in section 2.12 fusing the GPS information can improve the overall performance of the location estimation. All of these components are used in the Discrete Bayes Tracker – described in section 2.9.5 – to compute the final location estimation.

2.9.1 Grid Definition

I define a grid over the traversable area in the entire station. Generally speaking, traversable area is a subarea of the station where people and specially those with visual impairment can safely walk. Traversable area in the Diridon station is shown in Fig. 2.6. On the left side image you can see that there are two long and parallel pathways where the East pathway extends North to connect with a pedestrian tunnel under the main station. Travelers can cross the light rail tracks on two “crosswalks", with spring-operated gates. The grid is defined on East, North, Up (ENU) local tangent plane coordinates and has 440×153 cells that covers the whole Diridon station. The whole traversable area has been quantiszed with bin size of $1m$. The origin of the grid is in ENU coordinate and is located at far south west side of the VTA Diridon station. I obtained the geodetic coordinates of the grid's origin from Google Maps and transformed it to ENU coordinate. If a point in geodetic coordinate is to be represented in this grid, it must be transformed to ENU coordinate based on this origin and its corresponding cell is identified. Please refer to 2.7.1 section for more detail on the transformation methodology. X and y axes are oriented toward east and north respectively.

2.9.2 BLE Beacons RSSI Modeling

At the core of the self-localization problem is the idea of estimating state from sensor data (i.e. beacon RSSIs). For this specific problem, the state is the location of the phone in the transit station which is a set of quantities from beacons' data that are not accurately observable due to the noise in the RSSIs, but can be indirectly inferred. Probabilistic state estimation algorithms compute posterior distributions (a.k.a. belief distributions) over possible world states. This posterior is the probability distribution over the state $x(t)$ at time t, conditioned on measurements ($z(t)$. Biswas *et al.* [31] use the same approach for robot localization using WiFi strength signals in indoor environment. I model the received power from various beacons at each location x_j in the grid as a normally distributed random vector with independent entries (diagonal covariance). The mean of this random vector is set equal to the average signal vector $\{\overline{RSSI_j^i}\}$ computed from the measurements received within a square region with side of 5 meters (25 cells) centered at x_j (where i indexes the beacons). The entries of the diagonal covariance matrix are set to a constant value σ_B, as this was found to give more stable results than using the empirical variance values. (σ_B was set to 8 dBm in the experiments.) Thus, the conditional likelihood of the received signal can be expressed as:

$$p(\{RSSI^i\}|x_j) = \prod_i \frac{1}{\sqrt{2\pi}\sigma_B} e^{-\frac{(RSSI^i - \overline{RSSI_j^i})^2}{2\sigma_B^2}} \tag{2.12}$$

Given a measured vector $\{RSSI^i\}$, one can easily compute the posterior distribution over locations under uniform prior: $p(x_j|\{RSSI^i\}) \propto p(\{RSSI^i\}|x_j)$. The most likely location is computed as the cell x_j that maximizes this posterior distribution. Note that this approach involves computing $p(x_j|\{RSSI^i\})$ for all grid cells x_j.

In general, signal from only a limited set of beacons ($B(x_j)$) will be received at a given location x_j during fingerprinting. (Note that only if I receive at least 3 measurements from the same beacon within the region used to compute the average RSSI, will the beacon be included in Bx_j.) Let B be the set of beacons whose signal is received at run time. Care must be taken when computing $p(x_j|\{RSSI^i\})$ for any locations x_j such that the set of beacons with signal received during fingerprinting, Bx_j, does not match B. In this case, I adopt the following simple strategy: for each beacon in Bx_j that is not in B, the missing RSSI value in (2.12) is set to a small value (-95 dBm). Likewise, for each beacon in B that is not in

Bx_j, I add a “fake" beacon with the same small value for $\overline{RSSI^i_{x_j}}$ in (2.12).

2.9.3 GPS Error Modeling

Modern smartphones provide APIs that produce an estimate of the accuracy of GPS data in the form of radius of uncertainty. For example, in iOS, objects of the `CLLocation` class have a property named `horizontalAccuracy`, that represents the radius of uncertainty of the GPS measurements. In first approximation, this could be taken as the standard deviation σ_{GPS} of the GPS localization error. Root Mean Square (RMS), Distance Root Mean Square (DRMS), Twice Distance Root Mean Square (2DRMS), Circular Error Probability (CEP), 95% radius (R95), and Spherical Error Probable (SEP) are among the most common GPS error models used in navigation. These models assume that the distribution of GPS error is Gaussian. The GPS returns a location $(x, y)_{GPS}$ with a certain accuracy (radius of uncertainty). I assumed that, radius of uncertainty refers to the circular error probable, which is the radius of the circle where 50% of the mass is located. DMSE, 2DRMS, and R95 assume 63.213%, 98.169% and 95% as a percentile of the mass located in the circle. I assume that $p((x, y)|GPS)$ is an isotropic Gaussian (i.e. with co-variance matrix equal to $\text{diag}(\sigma_{GPS}, \sigma_{GPS})$), with mean value at $(x, y)_{GPS}$. Let $r(x, y)$ is the distance of a point (x, y) to $(x, y)_{GPS}$, and let $p(r|GPS)$ be its pdf. $p(r|GPS)$ turns out to be a Rayleigh density. The value of CEP is thus the median of $p(r|GPS)$, which is related to the σ_{GPS} of $p((x, y)|GPS)$ as by:

$$CEP = \sigma_{GPS}.\sqrt{2\log(2)} \approx \sigma_{GPS}.1.18 \tag{2.13}$$

Hence, I can assume that $\sigma_{GPS} = \textit{uncertainty radius}/1.18$. DMSE, 2DRMS, and R95 have the same rationale and their values can be calculated using

$\sigma_{GPS}.\sqrt{-2\ln(1-(F/100)}$, F is between 0% to 100%.

At this point $p((x,y)|GPS)$ is fully characterized and it is proportional to the likelihood $p(GPS|(x,y))$ under uniform prior on (x,y):

$$p(GPS|(x,y)) = R.p((x,y)|GPS) = R.\frac{1}{2\pi\sigma^2_{GPS}}e^{\frac{||(x,y)-(x,y)_{GPS}||^2}{2\sigma^2_{GPS}}} \tag{2.14}$$

where R is a constant that depends on σ. I observed, however, that the uncertainty radius produced by the API is not always reliable. Specifically, there are situations (like the one shown in Fig. 2.6, right), where the location provided by GPS may have a consistent bias that is poorly modeled by white additive Gaussian noise. For this reason, I decided to use a mixture model instead. Specifically, the probability $p(x_j|GPS)$ of being at a certain location x_j, where x_j is a cell in the grid, is modeled as a convex combination of a normal distribution, centered at the location reported by GPS and with a standard deviation equal to the uncertainty radius, and of a uniform distribution (the latter accounting for large deviations that may occasionally be expected) 2.15.

$$p(GPS|(x,y)) = \gamma_{GPS}\ .p(GPS|(x,y)) + (1-\gamma_{GPS}).K \tag{2.15}$$

where
$p(GPS|(x,y)) = \frac{1}{2\pi\sigma^2_{GPS}}e^{\frac{||(x,y)-(x,y)_{GPS}||^2}{2\sigma^2_{GPS}}}$

and K is a constant small number. I set γ_{GPS} to 0.5, σ_{GPS} to uncertainty radius, and K to 10^-4 in the mixture model.

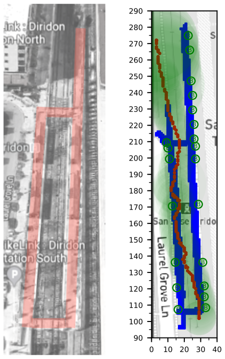


Figure 2.6: Left: An aerial picture of the San Jose Diridon light rail station, with the walkable areas highlighted. Right: A GPS track measured while walking on the East platform. Each point is shown with its uncertainty radius (displayed as a green transparent circle). Note the large localization error in the northernmost part of the track.

2.9.4 GPS and BLE Beacons Data Fusion

Let $p(RSSI|(x_j))$ or $p(GPS|x_j)$ the conditional density function (likelihood) of the received signal (RSSI or GPS) at a certain location x_j. The posterior probability $p(x_j|RSSI)$ or $p(x_j|GPS)$ represents the probability of being at a location x_j given the received signal. These quantities are related by the Bayes rule. E.g.,

$$p(x_j|RSSI) = \frac{p(RSSI|x_j).p(x_j)}{p(RSSI)} \tag{2.16}$$

where $p(x_j)$ is the prior probability of being at location x_j, which I assume to be uniform. $p(RSSI)$ is the prior probability of receiving a certain value of RSSI. Assuming conditional independence, the conditional density for both received signals can be expressed as follows:

$$p_{fus}(RSSI, GPS|x_j) = p(RSSI|x_j)p(GPS|x_j) \tag{2.17}$$

$$p_{fus}(x_j|RSSI, GPS) = \frac{p(RSSI|x_j, GPS).p(x_j|GPS)}{p(RSSI|GPS)}$$

Since RSSI and GPS are independent then:

$$p_{fus}(x_j|RSSI, GPS) = \frac{p(RSSI|x_j).p(x_j|GPS)}{p(RSSI)}$$

By expanding $p(RSSI|x_j)$ we get:

$$p_{fus}(x_j|RSSI, GPS) = \frac{p(x_j|RSSI).p(RSSI).p(x_j|GPS)}{p(x_j)p(RSSI)}$$

under uniform prior $p_(x_j)$, the posterior factorization will become:

$$p_{fus}(x_j|RSSI, GPS) = K.p(x_j|RSSI).p(x_j|GPS) \tag{2.18}$$

where K is a “normalizing” factor (which depends on RSSI and GPS) For multiple beacons I will also assume conditional independence:

$$p(RSSI^1, RSSI^2, ..., RSSI^i, GPS|x_j) = \prod_i p(RSSI^i|x_j).p(GPS|x_j) \tag{2.19}$$

Where i indexes the beacons. It is sometimes useful to allocate different “weights" to the two measurements being combined together. I enable this by expressing

(2.18) in the log domain and changing it to a convex combination:

$$\log p_{fus}(x_j|\{RSSI^i\}, GPS) \tag{2.20}$$

$$= (1-\alpha)\log p(x_j|\{RSSI^i\}) + \alpha \log p(x_j|GPS) + K$$

where $0 \leq \alpha \leq 1$ and K is a normalization constant. Note that smaller values of α assign more weight to the localization estimate from the BLE beacons, and vice-versa. In my experiments, I set $\alpha = 0.2$. my goal is to estimate the user's location x_j based on the received data (GPS and one or more RSSI values). I use the Maximum a posteriori (MAP) approach: by selecting the x_j that maximize the posterior probability:

$$x_{jMAP} = argmax\ p(x_j|RSSI, GPS) = argmax\ p(RSSI, GPS|x_j) \tag{2.21}$$

where the second equality holds under the uniform prior assumption. This shows that, under uniform prior, finding the MAP solution is equivalent to finding the maximum likelihood solution, i.e. $argmax\ p(RSSI, GPS|x_j)$.

2.9.5 Discrete Bayes Tracker

At the core of the self-localization problem is the idea of estimating state from sensor data (i.e. beacon RSSIs). For this specific problem, the state is the location of the phone in the transit station. The sensor data is a set of quantities from beacons' data that are not accurately observable due to the noise in the RSSIs, but can be indirectly inferred. Probabilistic state estimation algorithms compute posterior distributions (a.k.a. belief distributions) over possible world states. This posterior is the probability distribution over the state $x(t)$ at time t, conditioned on all past measurements ($z(t:)$ and all past actions $v(t:)$ that are displacements of the phone from one location state to another.

Various types of Bayes Filters have been introduced in the literature that are worth considering to solve the state estimation problem. The taxonomy of the important Bayes Filters follows:

- Bayes Filters:
    - Parametric:
        - Gaussian Filters:
            - Kalman Filter
            - Extended Kalman Filter
            - Information Filter
    - Nonparametric:
        - Histogram Filter:
            - Discrete Bayes Filter
        - Particle Filter

Gaussian Filters is a family of recursive state estimators. Gaussian Filters are categorized in the parametric filters, since they use parametric (moments or canonical) representation of Gaussian distribution to represent the belief distribution. All Gaussian Filters assume that the transition function and measurement (sensor observation or data) function are known. *Kalman Filter*(KL) and *Extended Kalman Filter* (EKF) are Gaussian Filters that represent the belief distribution by mean and covariance of the Gaussian distribution. The former assumes that the transition function and measurement functions are linear and the latter extends that to nonlinear transition and measurement functions by doing an approximate linearization technique using first order Taylor expansion. Information Filter is the same as KL and EKF except it represents the belief distribution by canonical representation of the Gaussian Filter thereby decreasing the computation cost. One drawback of Gaussian Filter for the self-localization problem is that one has to know the transition and measurement (beacon's RSSI behavior) models. In section 2.11.4, I show that range-based and power regression models do not provide a reliable model of RSSI behavior in noisy environments such as open area due to multipath fading and signal power variation over time. The other drawback is that the belief distribution is unimodal as it is finally represented by a Gaussian distribution. Since I am dealing with multiple RSSIs all over the deployment site, it would be unacceptable to obtain a peak of belief probability close the center of the site.

A popular alternative to Gaussian techniques are nonparametric filters. Nonparametric filters do not rely on a fixed functional form of the posterior, such as Gaussians. Instead, they approximate belief posteriors by a finite number of values, each roughly corresponding to a region in state space. Contrary to Gaussian Filters, nonparametric filters do not make strong parametric assumptions on the posterior distribution. They are well-suited to represent complex multimodal beliefs. For this reason, they are often the method of choice when one has to deal with global uncertainty, and when it faces hard data association problems that yield separate, distinct hypotheses.

Histogram filter is a nonparametric filter that first decomposes the state space into finitely many regions, and represents the belief posterior by a histogram. A histogram assigns to each region a single cumulative probability; they are best thought of as piecewise constant approximations to a continuous belief density. Discrete Bayes filters apply to problems with finite state spaces. That is, where the random variable $X(t)$ can take a finite number of values. Histogram filter can be used to discretize the state space thereby approximating the posterior for the original continuous state space. However, doing so imposes more computations and it is not of my interest as a discrete approximation of the belief distribution with some level of granularity ($1m^2$ in my case) suffices. That makes Discrete Bayes filters a suitable algorithm for solving the self-localization problem in my model.

In particle filter, on the other hand, the posteriors are represented by finitely many samples. The samples of a posterior distribution are called particles. There is no notion of belief update in particle filter. Particle filter is of the type of exploration and sampling techniques that samples the state space by certain sampling distribution so as to estimate the belief posterior distribution. The belief function from the previous time step or some combination of transition and measurement distributions may be used as sampling distribution.

In practice, the number of particles M is often a large number, e.g., $M = 1,000$. In some implementations M is a function of t or of other quantities related to the belief posterior at time t. The intuition behind particle filters is to approximate the belief posterior by the set of particles. Ideally, the likelihood for a state hypothesis $x(t)$ to be included in the particle set shall be proportional to its Bayes filter belief posterior. As a consequence, the denser a sub-region of the state space is populated by samples, the more likely it is that the true state falls into this region. This property holds only asymptotically for $M \to \infty$ for the standard particle filter algorithm. For finite M, particles are drawn from a slightly different distribution. In practice, this difference is negligible as long as the number of particles is not too small (e.g. $M100$) [54]. Hence, using particle filter would make sense if the sample size can justify the computation cost saving by avoiding to compute the posterior for all the state space. As I will show later in this section, by knowing the characteristics of the beacons' RSSIs, a good proportion of the beacons are not required to be computed in many of the states. So I do not calculate the belief posterior for location states in the nontravesable area. It turns out that the discrete Bayes filter is my best estimator among the mentioned state estimation techniques which possess an acceptable computational complexity. Later in chapter 3.1 particle filter is used. Another possible problem associated with particle filer is that the model shall be able to recover from errors induced from GPS and RSSIs as the estimated state may be far off the ground truth at time t but I need to give a chance to the model to correct itself soon afterwards. However, due to the constraints imposed by sampling distribution, the algorithm may get stuck in sampling from certain area that does not account for the location states where the corrected belief distribution has a higher probability. It turns out that the discrete Bayes filter is my best estimator among the mentioned state estimation techniques which possess an acceptable computational complexity. Nevertheless, that would not completely rule out the particle filter and its effectiveness on localization problem is worth a thorough investigation.

Due to noise and ambiguity (the same RSSI vector may be measured at different locations with similar likelihood), the distribution $p(x_j|\{RSSI^i\})$ (as well as the fused distribution) is often multimodal, with competing peaks that may lead to possibly large "jumps" in the estimated location. 2.7 shows the peaks (darker colors) in the two sides of the Diridon station while the actual location is on the west side (black circle).

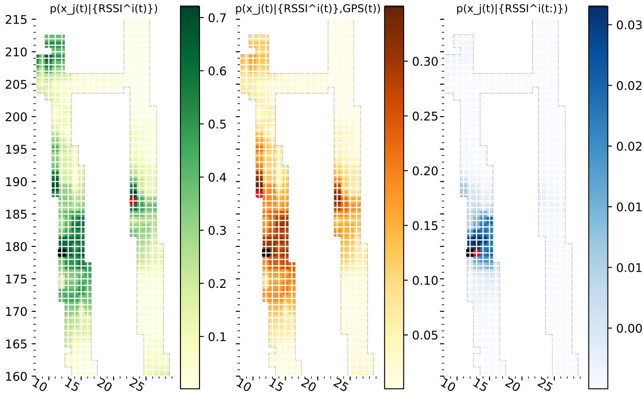


Figure 2.7: Example of posterior probability distributions of location over the grid at time t. Left: $p(x_j(t)|\{RSSI^i(t)\})$ (beacons only). Center: $p(x_j(t)|\{RSSI^i(t)\}, GPS(t))$ (GPS/beacons fusion). Right: $p(x_j(t)|\{RSSI^i(t:)\})$ (beacons with Bayesian tracker). Black circle is the actual location and red circle is the estimated location.

In order to overcome this effect, I employed a tracker, which smooths the computed trajectory based on a suitable dynamic prior. (Note that I don't apply the tracker to the location data from GPS, as it is normally already smoothed by the smartphone's API.) Although most recent work in the localization literature uses particle filtering trackers, I opted for a Discrete Bayesian Filter (sometime called Histogram Filter [54]) instead. This is a deterministic algorithm that is appropriate when the spatial domain is discrete (such as a grid) and the dynamic model is also discrete. In my case, I augment the "state" $x_j(t)$, representing the location of a person at time t, with the person's velocity $v(t)$. The velocity (more precisely, the displacement within one unit of time, which is assumed to be 1 second) can only take one of a small set of values. More specifically, I assume that from time $t-1$ to time t the user either remains within the same grid cell $x_j(t-1)$, or moves to one of the 8 neighboring cells. With some abuse of notation, I will write $x_j(t) = x_j(t-1) + v(t-1)$. The algorithm recursively recomputes the posterior distribution of location and velocity at each time under standard

Markovian assumptions as follows:

$$p(x_j(t), v_k(t)|\{RSSI^i(t:)\}) \propto p(\{RSSI^i(t)\}|x_j(t), v_k(t))\cdot$$
$$\sum_{\bar{j},\bar{k}} p(x_j(t), v_k(t)|x_{\bar{j}}(t-1), v_{\bar{k}}(t-1))p(x_{\bar{j}}(t-1), v_{\bar{k}}(t-1)|\{RSSI^i(t-1:)\})$$

where k indicates one of the possible 9 values of velocity, and $RSSI^i(t:)$ represents all RSSI readings up to and including time t.

I will assume that the RSSI readings are independent of the user's velocity, and that the user's velocity is independent of his or her location. Under these assumptions, the recursion becomes:

$$p(x_j(t), v_k(t)|\{RSSI^i(t:)\}) \propto p(\{RSSI^i(t)\}|x_j(t))\cdot$$
$$\sum_{\bar{j},\bar{k}} \delta(x_j(t) - (x_{\bar{j}}(t-1) + v_{\bar{k}}(t-1))p(v_k(t)|v_{\bar{k}}(t-1))$$
$$p(x_{\bar{j}}(t-1), v_{\bar{k}}(t-1)|\{RSSI^i(t-1:)\})$$

where $\delta(\cdot)$ is 1 when its argument is 0, 0 otherwise. For what concerns term $p(v_k(t)|v_{\bar{k}}(t-1))$, I will assume that with probability $1-\epsilon$ the velocity remains the same ($k = \bar{k}$), while with probability $\epsilon/8$ it may take any one of the other 8 possible values (where $0 \leq \epsilon \leq 1$ is a design parameter that was set to 0.2 in the experiments.) Note that, for each location x_j, the recursion only involves a small amount of operations, which are well manageable on a smartphone platform. The algorithm can be easily extended to the case of fused localization from GPS and BLE beacons.

2.10 Experiments and Results

The first section (2.11) of this chapter is dedicated to data collection, experiments, and results conducted with the goal of understanding the signal propagation behavior of the beacons in outdoor environment. Most studies used BLE beacons in indoor medium. I experimented with transmission and power regression models to test their capability of location estimation in Diridon station. The experiments were run in different conditions such as weather condition, presence or absence of crowd, beacon's orientation, beacon's height, angles between phone and beacon, etc. to find out the models' reliability in predicting the distance in the aforementioned conditions. My finding suggests that these models are not robust for a noisy and complex setup as is the case for Diridon station.

In the second section 2.12 of this chapter, I present the experiments I conducted at Diridon station to evaluate the stochastic model explained in section 2.9. The objective of these experiments are to understand to what extent the model can estimate the location when GPS and beacons are used in isolation as well as when their data are statistically fused.

I further investigated the possibility of reducing the number of beacons in Diridon station as proceeds in section 2.12.2.

2.11 Deterministic RSSI Models Evaluation

In this section I, first, examine the RSSI values collected for four different orientations over four hours total. Second, I explore the transmission and power regression model to obtain the distance of a phone with respect to a beacon to which I refer as distance throughout the text. Third, I investigate the relation between the orientation of the beacon and the received RSSIs. Lastly, I examine the proposed grid based fusion model tested at Diridon station. Receiver in this study is an iPhone 7 and transmitters are tough iBeacons from Kontakt company [55]. The transmission (TX) power of the beacons is set to 3 – ranging up to $20m$ – and the advertisement interval is set to 350 ms for the whole study. An iOS application was developed for data collection. RSSIs, time-stamps, and iPhone estimations at different positions and locations collected in an outdoor environment. The environment is subject to various noises due to radio frequency interference caused by surrounding people's phones or due to people who cross the space between the receiver and beacon among other reasons.

2.11.1 Analyzing RSSI Variations over Time and Space

In this section I examine how RSSIs vary at different angles and distances from beacon over an hour of data collection at 4 locations. Figure 2.8 shows schematic view of the data collection setup.

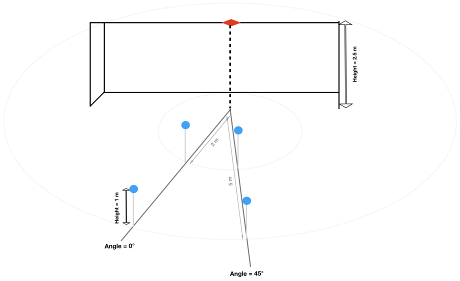


Figure 2.8: A schematic view of the data collection setup. The beacon is shown as a red diamond attached to a wall at $2.5m$ height. The receiver is located at $1m$ height on angle-distance combinations of $\{0°, 45°\}$ and $\{2m, 5m\}$ shown in blue circles.

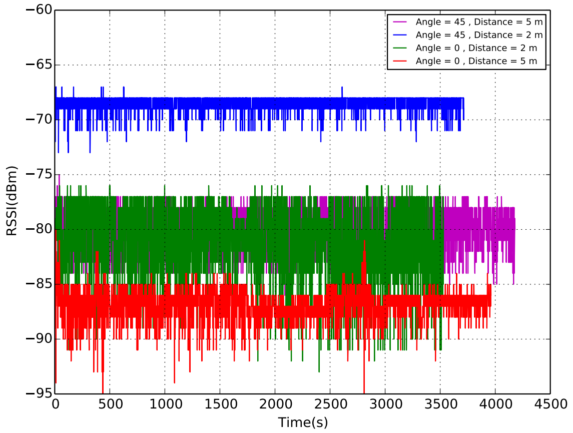


Figure 2.9: RSSIs collected over 1 hour at different locations of receiver as shown in figure 2.8.

Figure 2.9 shows 4 hours of RSSI collected at different locations of receiver as shown in figure 2.8. I did not observe a large variance in RSSI values over time (1 hour in this case). Furthermore, I found that there is a noticeable difference between receiving signal from beacon in different angles. For instance, at distance $2m$ I observed $\sim 12dBm$ RSSI difference between $0°$ and $45°$ angles which is well above the variance I get at one location over time.

2.11.2 Comparison of the Models in Distance Estimation

In this section, I examine the transmission and power regression model to estimate the distance between a receiver and beacon(s). I compare the resulting estimated distance with estimated distance that iPhone provides. iPhone can detect beacon(s) and provides a proximity estimation to detected beacon. The accuracy of the proximity value, measured in meters from the beacon, is what I call iPhone estimation hereafter. The goal of this experiment is to investigate the accuracy of transmission and power regression model in an outdoor environment and their robustness against different noises.

In this setup, I define $(O,\ B,\ L,\ r)$ as a setup combination where

- O is the orientation angle with respect to beacon and $O \in \{180°, 225°, 270°\}$,
- B is body orientation with respect to beacon and $B \in \{facing,\ non-facing\}$,
- L is the receiver location and $L \in \{hand,\ pocket\}$,
- $r\ (m)$ is distance and $r \in \{1,\ 2,\ 5,\ 10m\}$.

Figure 2.10 is a schematic view of the data collection setup.

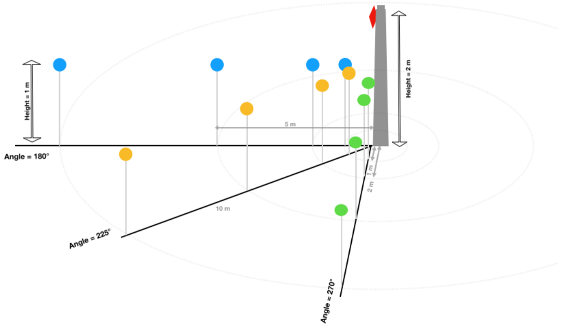


Figure 2.10: A schematic view of the data collection setup. The beacon is shown as a red diamond attached to a a concrete column at $2m$ height. The receiver is located at $1m$ height on all setup combinations of $(O,\ B,\ L,\ r)$.

I collected ∼180 measurements of RSSIs in 3 minutes for each setup combination $(O,\ B,\ L,\ r)$. Then, I categorized the data based on L and into three groups of holding the receiver in *hand*, putting the receiver in *pocket*, and *hand-pocket* (i.e. both). For each group, I compute the parameters n and x of the transmission model preceded in chapter 2.5 eq. (2.5). Measurements for n are shown in table 2.1. The noise parameter x is additive and its variance is independent of the distance [50]. To obtain the value of x, the variance of collected RSSIs, i.e. $\hat{p}_i$

, for each group is computed. I adopted power regression model to estimate the distance between the receiver and the beacon using eq. 2.11 described in chapter 2.5. I.e. $\hat{r_i} = A.(\frac{\hat{p_i}}{p_0})^B + C$.

I obtained three path loss components n and three constants of power regression for each group *hand*, *pocket*, and *hand-pocket*. There are 4229, 2032, and 6261 records in the 1st, 2nd and 3rd set, respectively. Table 2.2 shows the constant values A, B, and C for each group.

	Path Loss	**Hand-Pocket**	**Hand**	**Pocket**
	n	1.62	1.53	1.84

Table 2.1: Three path losses obtained for *hand*, *pocket*, and *hand-pocket* groups.

	Constants	**Hand-Pocket**	**Hand**	**Pocket**
	A	1.33582513	1.9445	0.5131481
	B	9.14734297	7.75377406	14.0193339
	C	0.65804331	0.03366047	1.23774671

Table 2.2: Three constants obtained for *hand*, *pocket*, and *hand-pocket* groups.

By knowing these parameters, one can compute path loss theoretical curves of transmission model and theoretical curves of power regression model. Theoretical curves of both transmission and power regression model are computed by incorporating the obtained path losses and constants into eqs. 2.5, 2.11. Figure 2.11 shows the distribution of RSSIs over computed real distances – distance between the receiver and beacon.

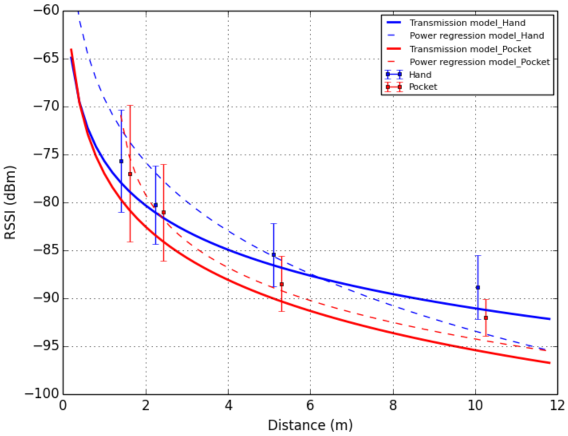


Figure 2.11: Comparison of theoretical curves of transmission model and power regression model, and empirical data. Colored dots and error bars showing the mean and standard deviation of RSSIs respectively vs. distance.

I use eq. (2.8) $\hat{r}_i = 10^{\frac{p_0 - \hat{p}_i}{10n}}.r_0$ where r_0 is 1 meter and p_0 is average of collected RSSIs at $r_0 = 1m$. $\hat{p}_i$ is received RSSI It is worth mentioning that the data I used for obtaining the path losses and constants are identical to the data I used for distance estimations. This makes the results very optimistic. Figures 2.12 and 2.13 show the mean and standard deviation of estimated distances, i.e. $\hat{r}_i$, compared to iPhone's estimations.

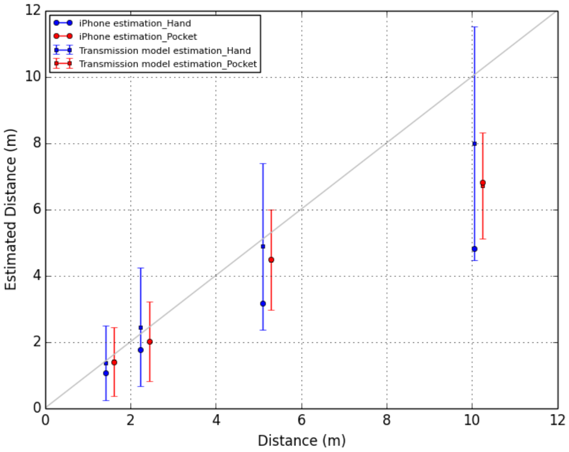


Figure 2.12: Comparison of estimated distances by means of transmission model and iPhone's estimations. Blue and Red colored dots and error bars show the mean and standard deviation of estimated distances for *hand* and *pocket* groups respectively. Blue and red circles show the iPhone's estimations. Gray line is the ground truth.

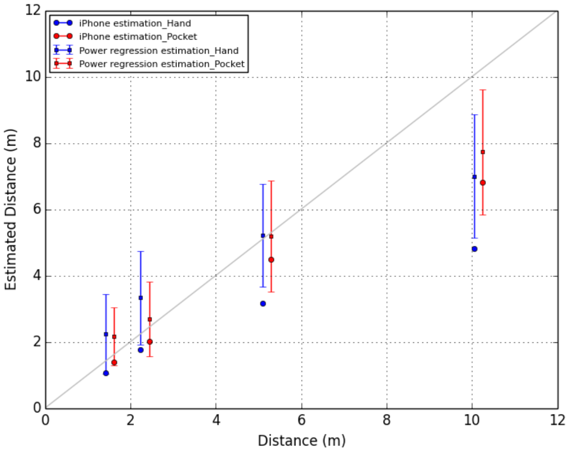


Figure 2.13: Comparison of estimated distances by means of power regression model and iPhone's estimations. Blue and Red colored dots and error bars show the mean and standard deviation of estimated distances for *hand* and *pocket* groups respectively. Blue and red circles show the iPhone's estimations. Gray line is the ground truth.

Notice that both transmission and power regression models yield estimated distances that are closer to the ground truth compared to iPhone estimations. Also, it turns out that the transmission model estimates the distances slightly better than the power regression model in this setting. However, I observed that, unlike Papamanthou *et al.* [50] observation, n were not in range of 2 to 4 for the outdoor mediums where I deployed the beacon in. I speculate that this is because of reflections, noises, and orientation of the beacon. I further explored how a beacon broadcasts signals based on its deployment orientation. I will discuss this later in this section.

2.11.3 Analyzing RSSI Variations based on Beacon's Orientation

Rezazadeh *et al.* [40] showed that vertical and horizontal positions of beacons affect their broadcasting range of the beacons in indoor environment. Here I study the positioning of a beacon to increase the overall coverage of the beacons in outdoor environment. To investigate the relation between RSSIs and the orientation of the beacon I collected $\sim$ 600 measurements of RSSIs in 10 minutes for each setup combination of $(O,\ B,\ L,\ r)$. I define $(O,\ B,\ L,\ r)$ as a setup combination where

- O is the orientation angle with respect to beacon and $O \in \{0°, 90°, 180°, 270°\}$,
- B is the body orientation with respect to beacon and $B \in \{facing\}$,
- L is the receiver location and $L \in \{hand,\ pocket\}$,
- $r\ (m)$ is distance and $r \in \{1,\ 2,\ 4,\ 6m\}$.

Figure 2.14 shows a schematic view of the data collection setup over 10 minutes at specified locations.

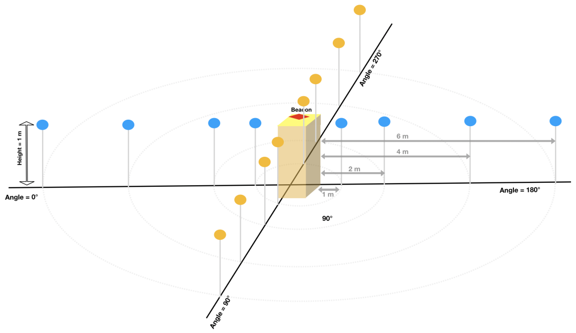


Figure 2.14: A schematic view of the data collection setup. The beacon is shown as a red diamond attached to a carton box at $1m$ height. The receiver is located at $1m$ height on all setup combinations of $(O,\ B,\ L,\ r)$ shown as blue and yellow circles. Antenna orientation was fixed to be horizontal. I.e. it is along the $0°/180°$ line. [2]

First, I obtained path loss n of transmission model and three constants of power regression model for the whole data-set. Notice that I have not categorized the data at this time.

I categorized the whole data based on O and into two groups of *horizontal* and *vertical*. Let's define the distance line as a line that crosses the centers of the receiver and the beacon and the antenna line as the line over which the beacon's omnidirectional antenna lays. The orientation is said to be *horizontal* if the distance line and the antenna line are the same. Whereas, the orientation is said to be *vertical* if the distance line is perpendicular to the antenna line [2].

Similar to previous experiment, I compute the theoretical curves of both transmission and power regression models. Figure 2.15 shows the distribution of RSSIs over computed real distances.

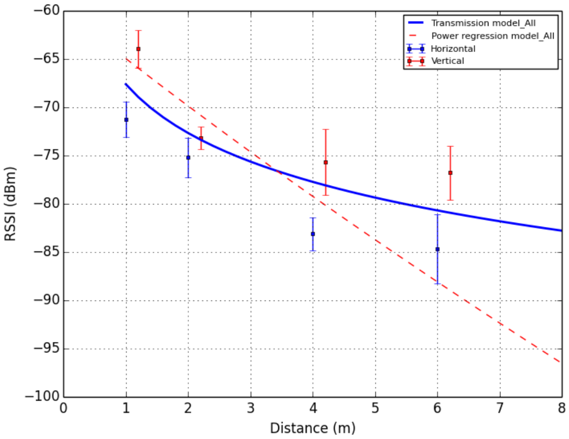


Figure 2.15: Comparison of theoretical curves of transmission model and power regression model, and empirical data. Colored dots and error bars show the mean and standard deviation of RSSIs respectively vs. distance

As I expected, I observed that the mean values of collected RSSIs in *vertical* group are $10dBm$ higher than collected RSSIs in *horizontal* group. I use eq. (2.8) $\hat{r_i} = 10^{\frac{p_0 - \hat{p_i}}{10n}}.r_0$ where r_0 is 1 meter and p_0 is average of collected RSSIs at $r_0 = 1m$ meter. $\hat{p_i}$ is received RSSI. Figures 2.16 and 2.17 show the mean and standard deviation of estimated distances, i.e. $\hat{r_i}$, compared to iPhone's estimations.

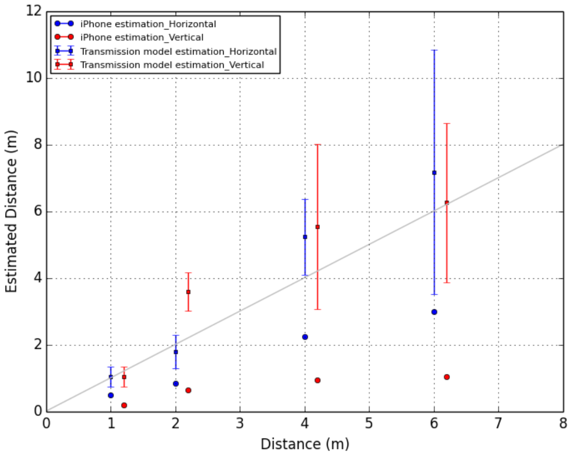


Figure 2.16: Comparison of transmission model estimated distances and iPhone's estimations. Blue and Red colored dots and error bars show the mean and standard deviation of estimated distances for *horizontal* and *vertical* groups respectively. Blue and red circles show the iPhone's estimations. Gray line is the ground truth.

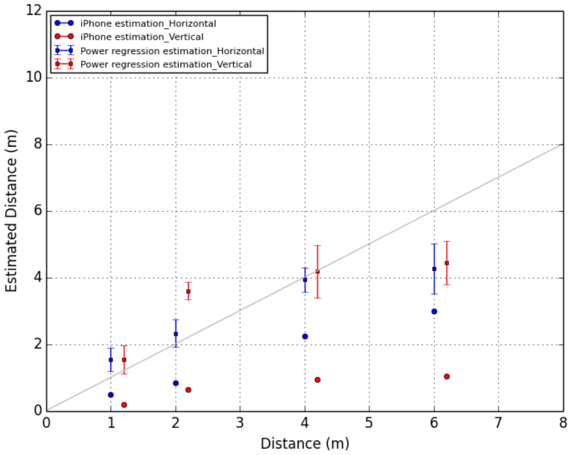


Figure 2.17: Comparison of power regression model estimated distances and iPhone's estimations. Blue and Red colored dots and error bars show the mean and standard deviation of estimated distances for *horizontal* and *vertical* groups respectively. Blue and red circles show the iPhone's estimations. Gray line is the ground truth.

Notice that both transmission and power regression models yield estimated distances that are closer to the ground truth compared to iPhone estimations. Also, it turns out that the transmission model estimates the distances slightly better than the power regression model.

2.11.4 Data Collection from two Beacons at Diridon Station

In this section, I conducted preliminary data collection at Diridon station. With the help of IBM and VTA we deployed 21 iBeacons at VTA Diridon station. The goal of this data collection is to investigate the accuracy of distance estimation using the two transmission and power regression models in the presence of the environmental noise in the station. I collected data from 2 beacons with $\sim 8m$ distance from one another to investigate the RSSIs and noise in the environment when estimating the distance of a receiver from the beacons. Figure 2.18 shows a schematic view of data collection setup. I collected data for each setup combination of $(L,\ r)$ from two beacons. Here, I define $(L,\ r)$ as a setup combination where

- P is the data collection path and $L \in \{path0,\ path1,\ path2\}$,
- $r\ (m)$ is the distance between the receiver and a beacon. $r \in \{0,\ 2,\ 4,\ 8m\}$ is the distance between receiver and beacon 1, and $r \in \{8,\ 10,\ 12,\ 16m\}$ is the distance between receiver and beacon 2. The receiver was held in hand and the body orientation of the receiver was facing towards the beacon 1 or 2 throughout the data collection.

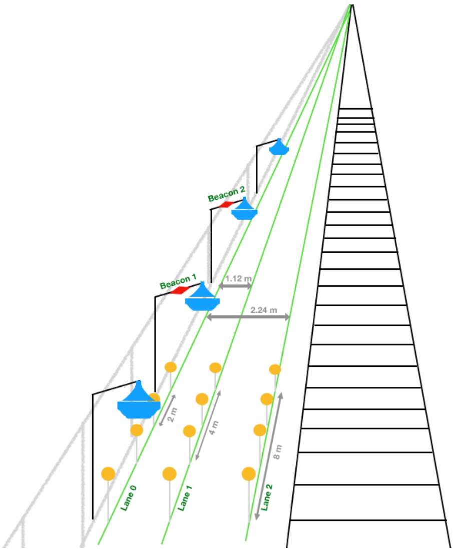


Figure 2.18: A schematic view of the data collection setup. The beacons are shown as a red diamond attached to the lamps at $3.56m$ height. The receiver is located at $1m$ height on all setup combinations of $(L,\ r)$ shown as yellow circles.

First, I obtained path loss n of transmission model and three constants of power regression model for the whole dataset. Notice that I have not categorized the data at this time.

Then, I categorized the data based on P and into three groups of collected data in *path 0*, *path 1*, and *path 2*. Similar to previous experiments I compute the theoretical curves of both transmission and power regression models. Figure 2.19 shows the distribution of RSSIs over computed real distances.

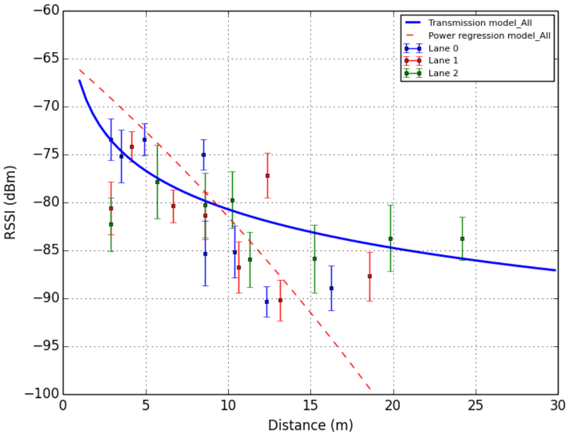


Figure 2.19: Comparison of theoretical curves of transmission model and power regression model, and empirical data collected at Diridon station (all three paths included). Colored dots and error bars show the mean and standard deviation of RSSIs respectively vs. distance

I observe that the mean values of collected RSSIs do not quite follow the theoretical curves of transmission model and power regression model. Another observation was that mean RSSIs increase as I move further away from the beacons in distances more than $15m$ contrary to my expectation to receive lower RSSIs in higher distances. Perhaps, this is because of the reflection of the beacons' signals when they hit the ground which manifests itself in larger distances. I also observed that at $8m$ distance from the 1st and the 2nd beacons, I received mean RSSI of $-85dBm$ and $-75dBm$ respectively despite the fact that the distances are the same. This suggests the presence of noise in the environment that arises from environmental noises or inter-beacons signal interference.

I use eq. (2.8). To recall

$\hat{r_i} = 10^{\frac{p_0-\hat{p_i}}{10n}}.r_0$

where r_0 is $2.16m$ and p_0 is the average of collected RSSIs at $r_0 = 2.16m$ meter.

$\hat{p}_i$ is the received RSSI. Figure 2.21 shows the comparison between estimated distances, i.e. $\hat{r}_i$, using transmission and power regression models, and iPhone estimation.

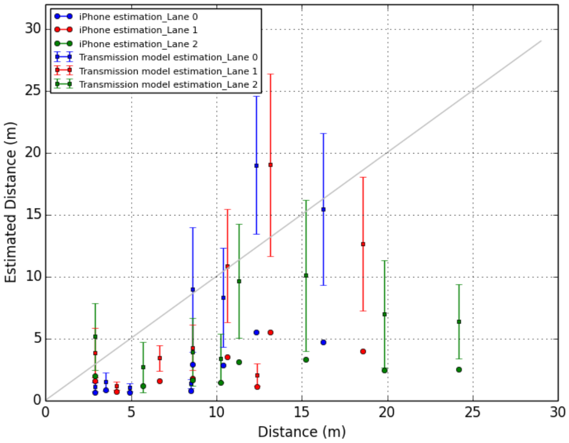


Figure 2.20: Comparison of transmission model estimated distances and iPhone's estimations. Blue, red, and green colored dots and error bars showing the mean and standard deviation of estimated distances for *path 0*, *path 1*, and *path 2* groups respectively. Blue, red, and green circles show the iPhone's estimations. Gray line is the ground truth.

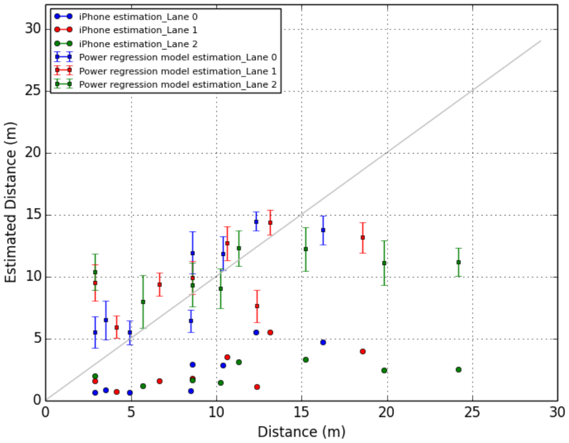


Figure 2.21: Comparison of power regression model estimated distances and iPhone's estimations. Blue, red, and green colored dots and error bars showing the mean and standard deviation of estimated distances for *path 0*, *path 1*, and *path 2* groups respectively. Blue, red, and green circles show the iPhone's estimations. Gray line is the ground truth.

Notice that both transmission and power regression models yield estimated distances that are closer to the ground truth compared to iPhone estimations. The standard deviation of estimated distance using transmission model is higher than power regression model. Also, it turns out that the power regression model estimates the distances slightly better than the transmission model. Recall that data I used for obtaining the path losses and constants are identical to the data I used for distance estimations. This makes the results very optimistic. I observed that to obtain the coefficients of both transmission and power regression models, one need to train the model based with enough samples at one position. This is not feasible in large outdoor environment with the presence of dynamic changes (i.e weather condition, signal reflections, presence of travelers and trains).

2.12 Proposed Stochastic Model Experiments

The idea of this proposed model is to deal with all the uncertainties I mentioned earlier using the two deterministic models. The proposed stochastic model fuses data from GPS and beacons and estimate the location of the user in a discrete state space – grid over traversable area. Earlier in section 2.9.3 I briefly mentioned that GPS error in Diridon station is not precise enough to route a person with visual impairments. Figure 2.22 is the aerial image from Google map. I walked on a path toward south and recorded the actual and GPS locations. Collected GPS shows how inaccurate the GPS is at the station.

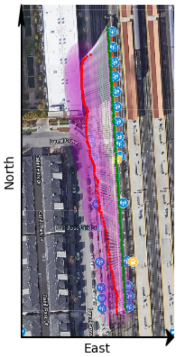


Figure 2.22: Received GPS locations at the station. Each green circle shows the ground truth location at time t along a path. Each red circle is the GPS location at time t. Each transparent purple circle shows the GPS location uncertainty at time t that ranges from 10 to 65 meters. Each white line shows the distance between a ground truth location and GPS estimated location at time t. The aerial image of the station has been taken from Google Map.

To improve the GPS location estimation we deployed 21 tough iBeacons at the station above the ground. TX power of each beacon is set to $3(-12dBm)$ that can transmit signals up to 20 meters. Advertisement interval is set to $350ms$ that broadcasts ~ 3 signals every second. Beacons are deployed on each lamp located at the station. The height of lamps are $\sim 3.8m$. The distance between lamps vary.

2.12.1 Data Collection

I collected data from 21 BLE beacons and GPS. The beacon dataset is a set of records each comprises time-stamp, beacon's minor ID, and RSSIs at time t. The GPS dataset is a set of records each comprises time-stamp, latitude, longitude, altitude, and GPS location uncertainty at time t. I collected the mentioned dataset for each setup combination of $(L,\ D,\ V)$ where

- P is the path along which I walked and $L \in \{path0,\ path1,\ path2,\ path3,\ path4,\ path5,\ path6,\ path7,\ path8,\ path9,\ path10\}$,
- D is the moving direction and $D \in \{north, south\}$ for path 0 to 4, and $D \in \{east, west\}$ for path 5 to 10,
- V is the velocity of walking along a path and $V \in \{\sim 0.5m/s,\ \sim 1m/s\}$.

The phone was held $1m$ high in my hand throughout the data collection. Fingerprinting was performed in January of 2019 from RSSI data collected with an iPhone 7. I walked each path P towards $D = north$ and $D = south$ with an approximate constant velocity of $V =\sim 0.5m/s$. The geodesic coordinates of the endpoint locations of each path were measured using Google Maps, and transformed to grid coordinates. By timestamping the start and end of the walk, the walking velocity was measured, which allowed me to assign a timestamp to each cell in the grid overlapping with the path, and thus to record RSSI measurements for that cell. Overall, data was collected from walking over 22 paths (including walking on the same path in opposite directions). Figure 2.23 shows the image of the station and the paths along which I walked for the data collection. There are 3 paths in the east side, 2 paths in the west side, 3 paths in the north cross walk, and 3 paths in the south cross walk of the platform. I used the ($L,\ D,\ V =\sim 0.5m/s$) data for training. It is worth mentioning that, in figure 2.23 blue circles show the 21 beacons locations, deployed at Diridon station. Two red circles are defective beacons that stopped working. The one that is closer to the north stopped working before the first data collection and the one that is closer to the south stopped working after first trial of data collection.

In addition, I mimicked a case with fewer (8) BLE beacons available (see layout of this subset of beacons in Fig. 2.27, last two plots), by considering data measured only from these beacons.

I tested the performance of the model over three different sets of trials to account for different weather conditions in the dataset. The first set (*Trial Set 1*) was collected on the same day of fingerprinting. For trial 1 data collection I used the (L, D, V=1m/s) setup. Similar setup to training data with velocity of $\sim 1m$. Overall, data was collected from walking over 10 paths (path 0 to 5 toward north and south directions). This trial set represents an “ideal" situation, as fingerprinting and data collection were conducted under identical conditions.

Timestamped RSSI data was collected during the trials. By measuring the location of the endpoints of these paths, and recording the start and end time of each walk, I was able to estimate the “ground truth" location at all times and thus to precisely measure the localization error. Ground truth locations are calculated as follows: Given the length of each path, the velocity V, the starting and ending geodetic location of the walking path, and time stamps, I compute the ground truth location (x_{GT}, y_{GT}) at each time-stamp t. First I convert geodetic coordinates (latitude and longitude) of starting and ending points to ENU coordinate system with respect to the grid origin I call them $(x, y)_s$ and

$(x, y)_e$ respectively. Then the angle $(angle)$ between $(x, y)_s$ and $(x, y)_e$ is measured by calculating the arctan $(y_e - y_s)/(x_e - x_s)$ between two points. Knowing timestamp (t), velocity (V), I can compute the distance via $dist = V \times t$. Given $dist$, $angle$, and $((x, y)_s)$, ground truth location at time t on the grid is calculated using
$y_{GT} = (sin(angle).dist) + y_s$ and
$x_{GT} = (cos(angle).dist) + x_s$.

A second smaller set (*Trial Set 2*) was collected in May of 2019, 4 months after fingerprinting. Of note, all beacons were turned off for a period of time (using the Kontact's beacon management app) between Trial Set 1 and 2, then turned on again before the data collection for Trial Set 2 began. One of the beacons (located on the East platform) stopped working in the process, and thus only 20 beacons were available for Trial Set 2. 4 trials were conducted in an identical fashion as for the previous case, with me walking over straight paths with known endpoints. Path reconstruction was also conducted with the reduced set of beacons (only 7 beacons, due to the aforementioned beacon failure).

The last set (Trial Set 3, collected in July of 2019) is comprised of 3 trials with me walking through paths that included multiple turns (crossing the crosswalks and walking in two east and west side of the platform). For these trials I don't have ground truth measurements of the experiment's location at all times, and thus cannot compute the localization error. I walked with different speeds from $0m/sto1.5m/s$ However, I recorded the times at which I took each turn, which allowed me to associate each data point with the segment (between two consecutive turns) when that data was collected. Based on this information, I was able to compute all "jumps" – situations in which the system returns a location that is in an incorrect path segment. The error metric in this case is the proportion of the "jump" events within a path.

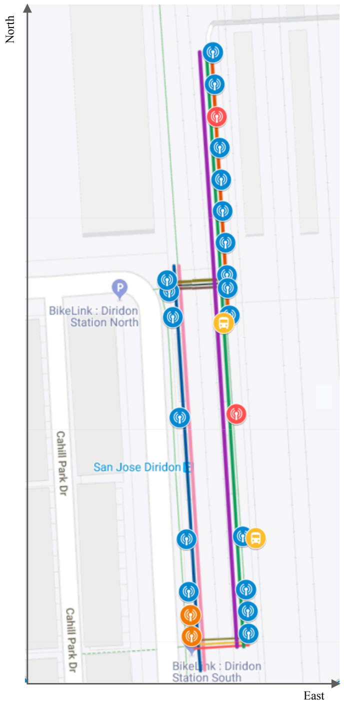

North
East
BikeLink : Diridon Station North
San Jose Diridon
Cahill Park Dr
Cahill Park Dr
BikeLink : Diridon Station South

Figure 2.23: Circles show the 21 beacons locations, deployed at Diridon station. *path 0* is shown in red and is very close to a thick wall. Path 0 is also under the lamps on the east side of the platform (see figure 2.18). *Path 1* is shown in green and is in the middle of the east side of the platform. *Path 2* is shown in purple and is very close to the track on the east side. *Path 3* is shown in pink and is very close to the track on the west side of the platform. *Path 4* is shown in light blue color and is in the middle of west side of the platform. Paths 5 ,6, 7 is on the north crosswalk and paths 8, 9, 10 are on the south crosswalk. The distance between each path is approximately $1m$

Figure 2.24 visualizes the RSSI signatures based on the data collected in path 1 on the east side. Each column corresponds to one beacon. Each row represents a location at which I collected the beacons' RSSI. I.e, a row at, say, $y = 25$ in the graph contains the RSSI of all beacons received at $X = 25$ meters. Higher RSSIs are darker. In other words, each row is a RSSI signature. Each column represents the RSSI measured from that beacon. Walking down a column from $0m$ to $175m$, we see the RSSI values measured from that beacon as I walk from $X = 0$ to $X = 175$.

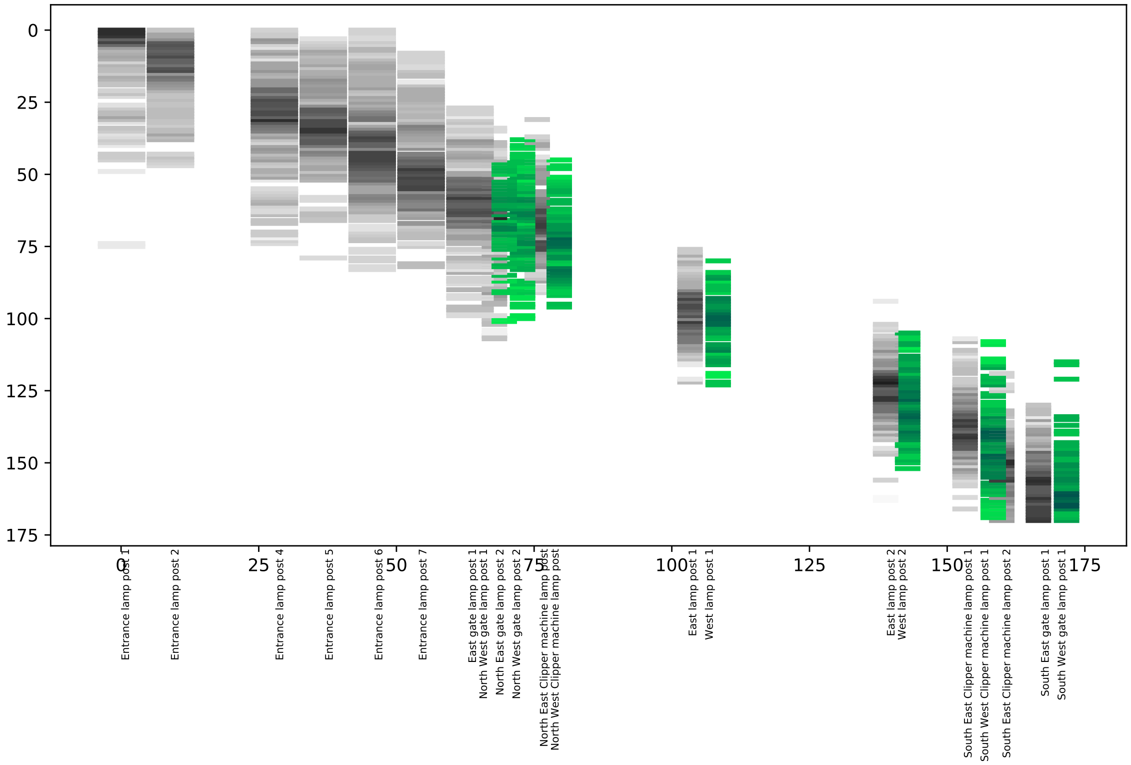


Figure 2.24: RSSI signatures on the east side of the platform. Each column corresponds to one beacon. Each row represents a location at which I collected the beacons' RSSI. Green rectangles are RSSIs received on the east side from beacons from the west side of the platform.

The process for location estimation starts with training the model as defined in eq. 2.19 with the training dataset and testing the model with testing dataset.

In training phase, I calculate the $RSSI^i_{mean}(x, y)$ for each cell in the grid with respect to each beacon i over all the $RSSI$ values in the cell and its surrounding cells. That is, for each cell x, y I calculate $RSSI_{mean}(x, y)$ based on $RSSI$s in cells

(i, j) where $i \in \{x{-}2,\ x{-}1,\ x,\ x{+}1,\ x{+}2\}$ and $j \in \{y{-}2,\ y{-}1,\ y,\ y{+}1,\ y{+}2\}$ if such cell exists in travesrable area. In other words, $RSSI_{mean}(x, y)$ computed from the measurements received within a square region with side of 5 meters (25 cells). Figure 2.25 illustrates how $RSSI_{mean}(x, y)$ is calculated from 25 surrounding cells.

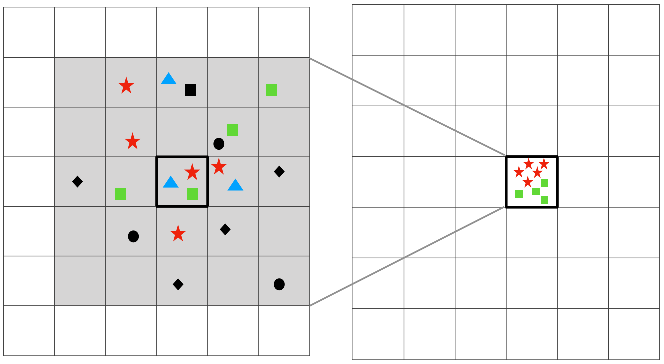


Figure 2.25: An illustration of calculating $RSSI^{i}_{mean}(x, y)$. Grey cells are 25 surrounding cells around (x, y) cell (shown with a thicker border). Each symbol (star, square, ...) corresponds to the beacon source of a received RSSI. RSSI values have not been represented in the illustration. Cell (x, y) received 3 RSSIs from 3 beacons during training data collection. 3 RSSIs are visualized as star, square, and triangle in the cell. In order to compute $RSSI^{i}_{mean}(x, y)$ I check the 25 surrounding cells and count the star, square, and triangle. If there are more than 3 of that shape in the surrounding cells, then I add them to cell (x, y). At the end we have 5 stars and 4 squares. Say the 5 star RSSIs have been received from beacon b_1 and 4 square RSSIs from another beacon b_2. Now, mean value of b_1, and b_2 RSSIs can be calculated for cell (x, y). Cell (x, y) consists of two mean values, $RSSI^{1}_{mean}$ and $RSSI^{2}_{mean}$.

Figure 2.26 reads similar to figure 2.24, but visualizes the calculated $RSSI^{i}_{mean}(x, y)$ instead of RSSIs. Each row is the computed $RSSI^{i}_{mean}$ of RSSIs I received from beacons during training phase. Each column represents the $RSSI_{mean}$ measured from that beacon. Walking down a column from $0m$ to $175m$, we see the $RSSI_{mean}$ values measured from that beacon as I walk from $X = 0$ to $X = 175$.

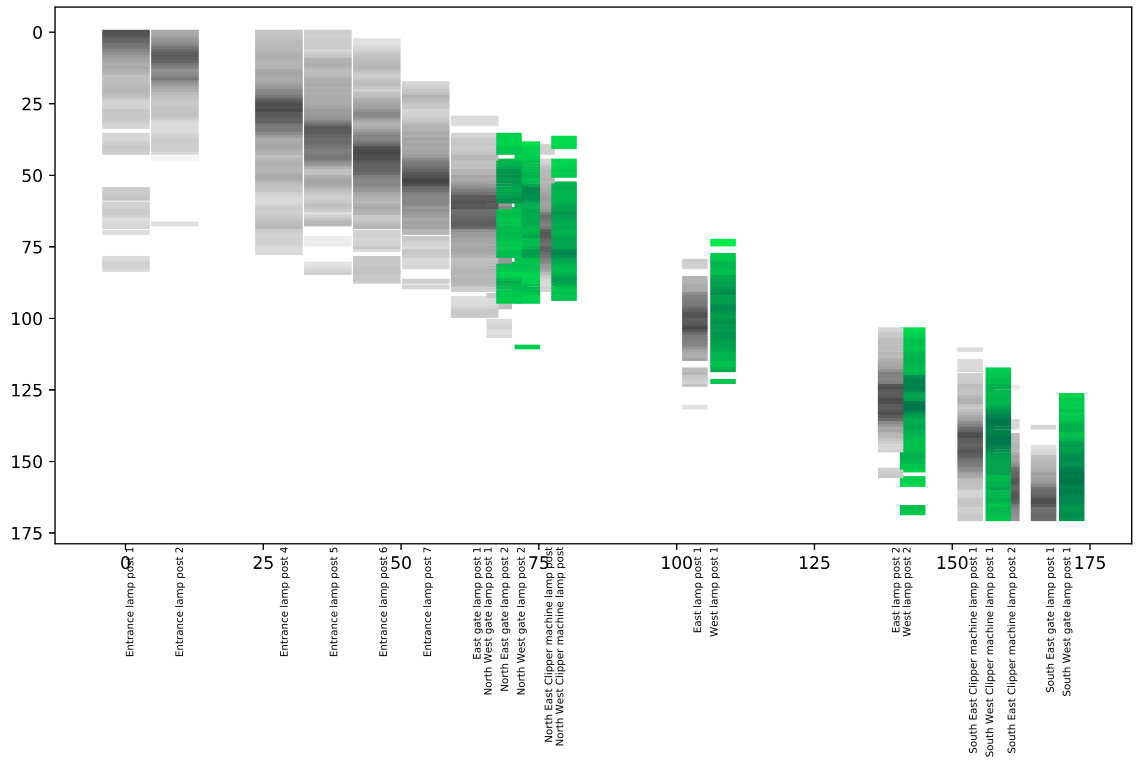


Figure 2.26: $RSSI^i_{mean}$ on the east side of the platform based on walking on path 1. Each column corresponds to one beacon. Each row represents a location at which I measured $RSSI^i_{mean}$. Green rectangles are computed $RSSI^i_{mean}$ during training phase on the east side from beacons from the west side of the platform.

2.12.2 Results

I tested multiple configurations of the system: GPS tracks, BLE beacons tracks (Sec. 2.9.2), and GPS/BLE beacon fused tracks (Sec. 2.9.4). In addition, for the last two modalities, I experimented with the use of the Bayes discrete filter tracker (Sec. 2.9.5). I experimented with all 21 beacons, as well as with the reduced set of 8 beacons. I have run the model 13860 ($5 \times 9 \times 4 \times 11 \times 7$) times by using all possible combinations of the following parameters' value sets. $\alpha \in \{0,\ 0.2,\ 0.5,\ 0.7,\ 1.0\}$. $\sigma \in \{2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10\}$. $N/A-RSSI_{mean}(x,y) \in$

$\{-95,\ -100,\ -115,\ -130\}$ (dBm).

$\epsilon \in \{0,\ 0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9,\ 1.0\}$.

$\gamma \in \{0.001,\ 0.01,\ 0.1, .2, .3, .4, .5\}$. I observed that the combination of $\alpha = 0.2$, $\sigma = 8.0$, $\epsilon = 0.2$, $\gamma = 0.3$ and $N/A - RSSI_{mean}(x, y) = -95dBm$ results in more accurate location estimations. I report the root mean square distance between the ground truth location at time t and the estimated location as reported by the system based on the RSSI vector and GPS location collected at that point. Root mean squared distance error eq. 2.22 is used to compute the error of the model for trail set 1 and 2. For trial set 3 due to in availability of ground truth locations I report proportion of the “jump" events within a path.

$$rmse = \sqrt{(y_{est.} - y_{GT})^2 + (x_{est.} - x_{GT})^2} \tag{2.22}$$

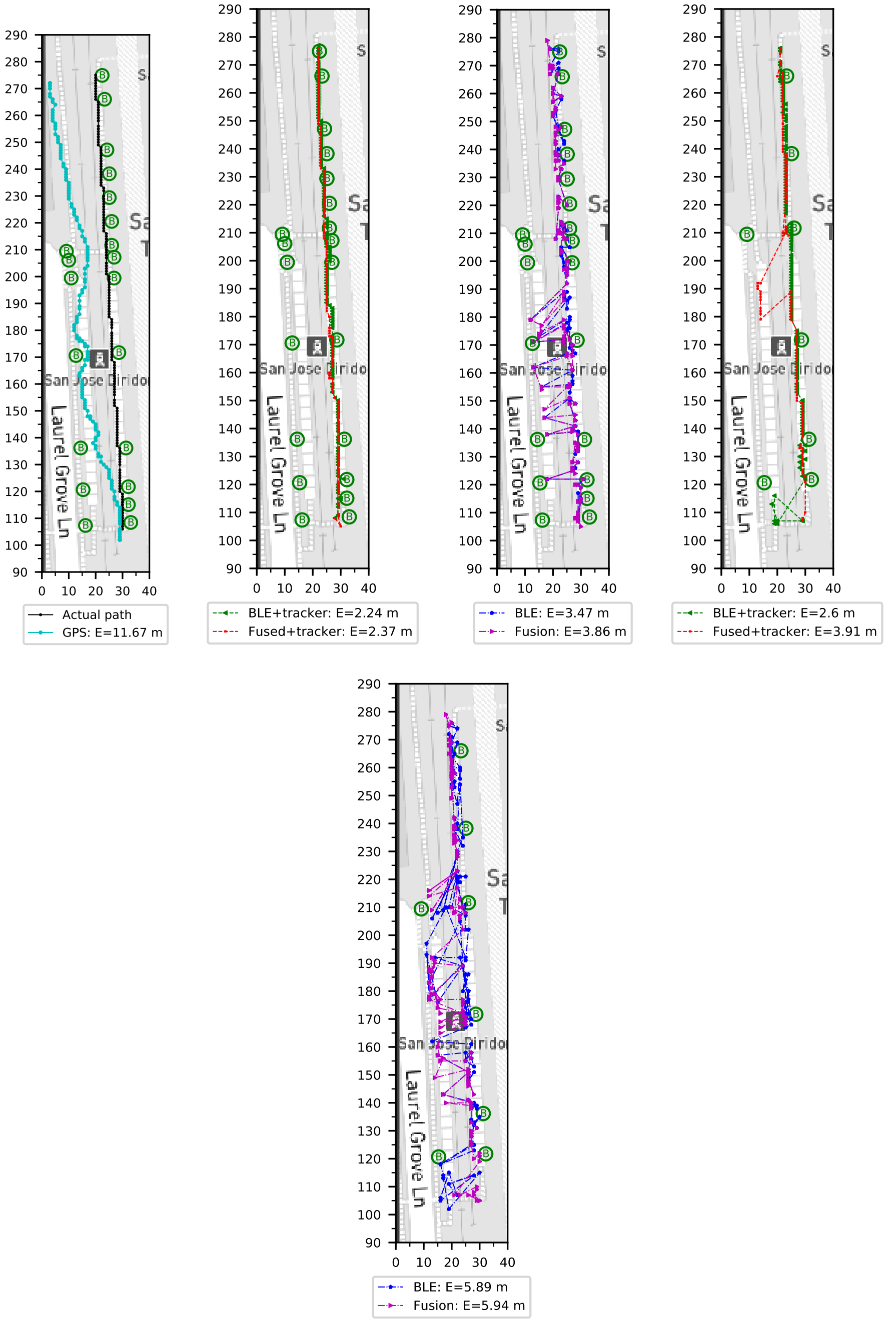


Figure 2.27: Experiments with a sample path (East platform) from Trial Set 1. The left plot shows the actual path taken (black) with the track estimated from GPS (light blue) and the average error computed for this path. The next two plots show results using all 21 beacons (whose locations are shown on the map), while the last two plots only 8 beacons are used (locations also shown).

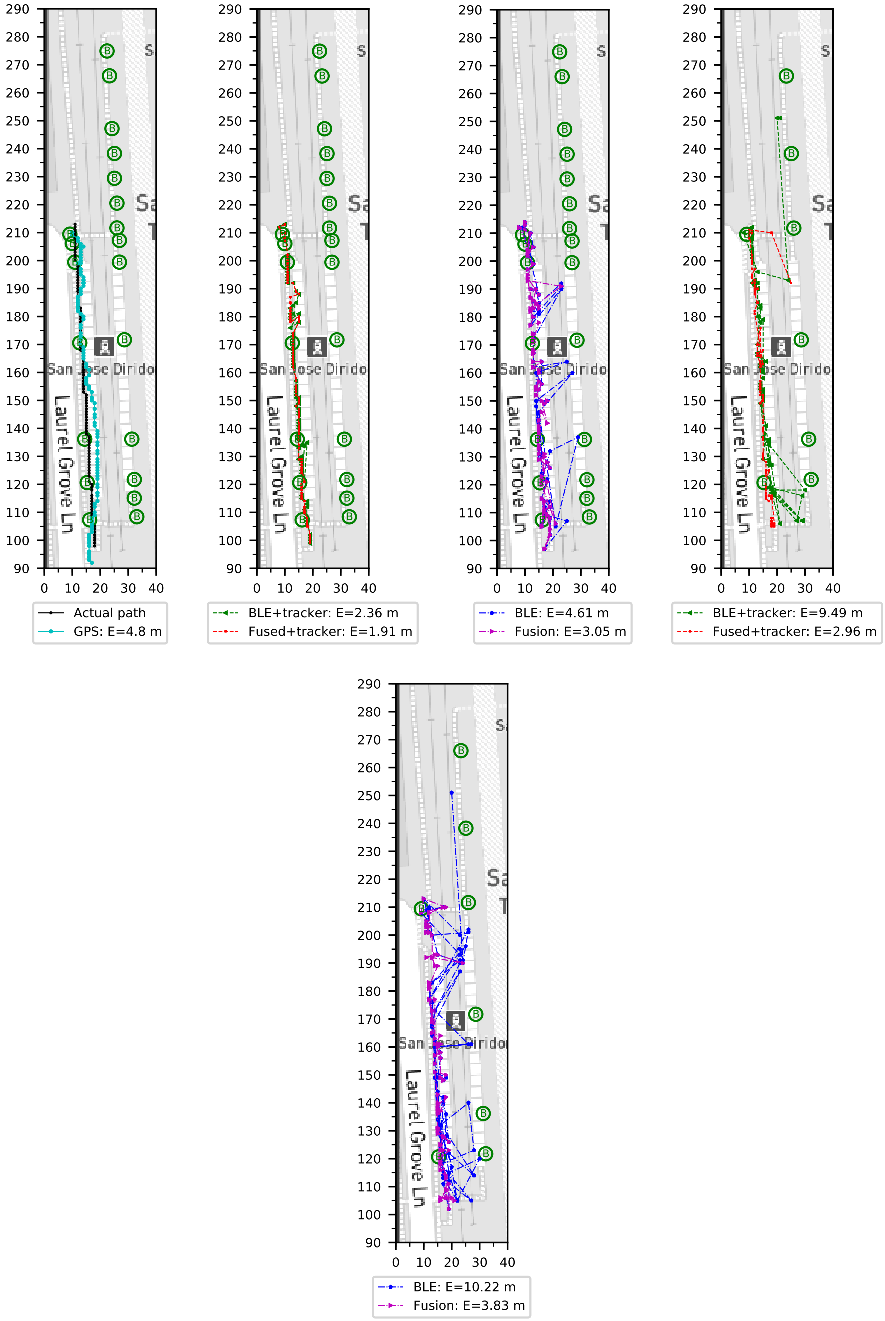


Figure 2.28: Experiments with a sample path (West platform) from Trial Set 1. See caption of Fig. 2.27.

Localization errors, averaged over all trials for each trial set, are shown in Tab. 2.3 using the chosen metrics. Figs. 2.27–2.31 show tracks computed for representative individual paths. A common characteristic of all trial sets is that the GPS tracks were for the most part correct while walking on the West platform, but grossly incorrect when walking on the East platform. This results in the large measured average error reported for the GPS tracks. Localization using only BLE beacons also produces substantial error, especially for the portion of the platforms facing each other. This is due to the fact that, when standing on one platform, the distance to one or more beacons in the other platform is often shorter than the distance to the nearest beacons on the same platform. This generates a multimodal posterior distribution, which results in frequent “jumps" from one platform to the other. In fact, beacon-based localization generates errors also when walking on the West platform, where GPS produces very good results. The situation was aggravated by the fact that, as noted above, one beacon on the East platform stopped functioning after Trial Set 1. In general, when using fewer beacons (see Fig. 2.30), the error increases, as expected.

Full set	**GPS**	**BLE**	**Fusion**	**BLE+tracker**	**Fusion+tracker**
Trial Set 1	10.38	4.36	3.83	2.72	2.55
Trial Set 2	7.61	6.65	5.18	6.33	5.81
Trial Set 3	29%	17%	19%	9%	14%

Reduced set	**GPS**	**BLE**	**Fusion**	**BLE+tracker**	**Fusion+tracker**
Trial Set 1	10.38	8.10	5.40	5.37	4.24
Trial Set 2	7.61	11.04	6.01	8.72	6.83
Trial Set 3	29%	24%	25%	11%	21%

Table 2.3: Error computed over all trials in each trial set. The error is expressed as root mean square distance between estimated and actual location for Trial Sets 1 and 2, and as the “jump proportion" for Trial Set 3. Top: all available beacons used (21 for Trial Set 1, 20 for Trial Sets 2 and 3). Bottom: reduced set of beacons used (8 for Trial Set 1, 7 for Trial Sets 2 and 3).

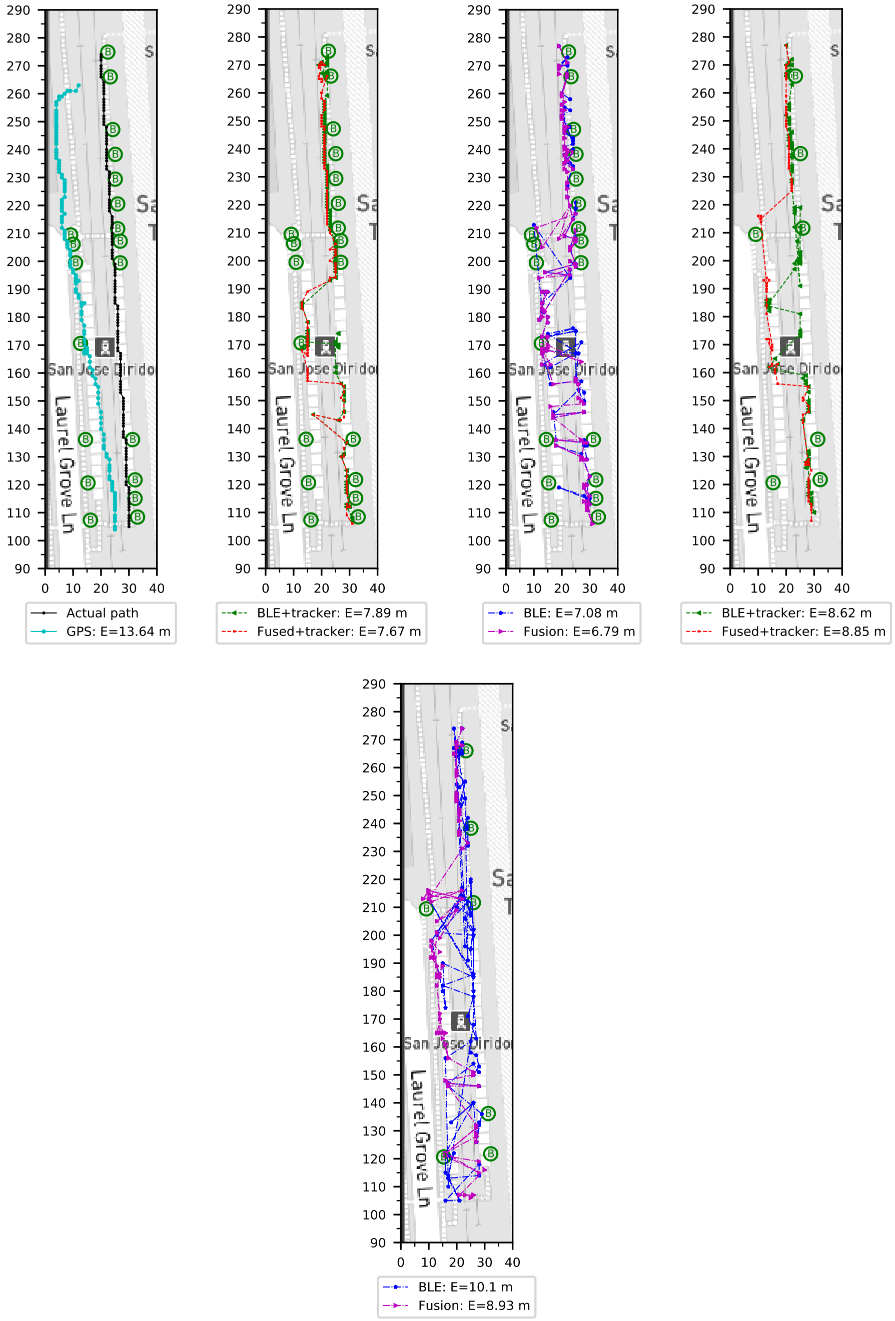


Figure 2.29: Experiments with a sample path (East platform) from Trial Set 2. See caption of Fig. 2.27. Note that, due to a beacon failure, there were only 20 beacons available, of which a subset of 7 beacons was used for the last two plots.

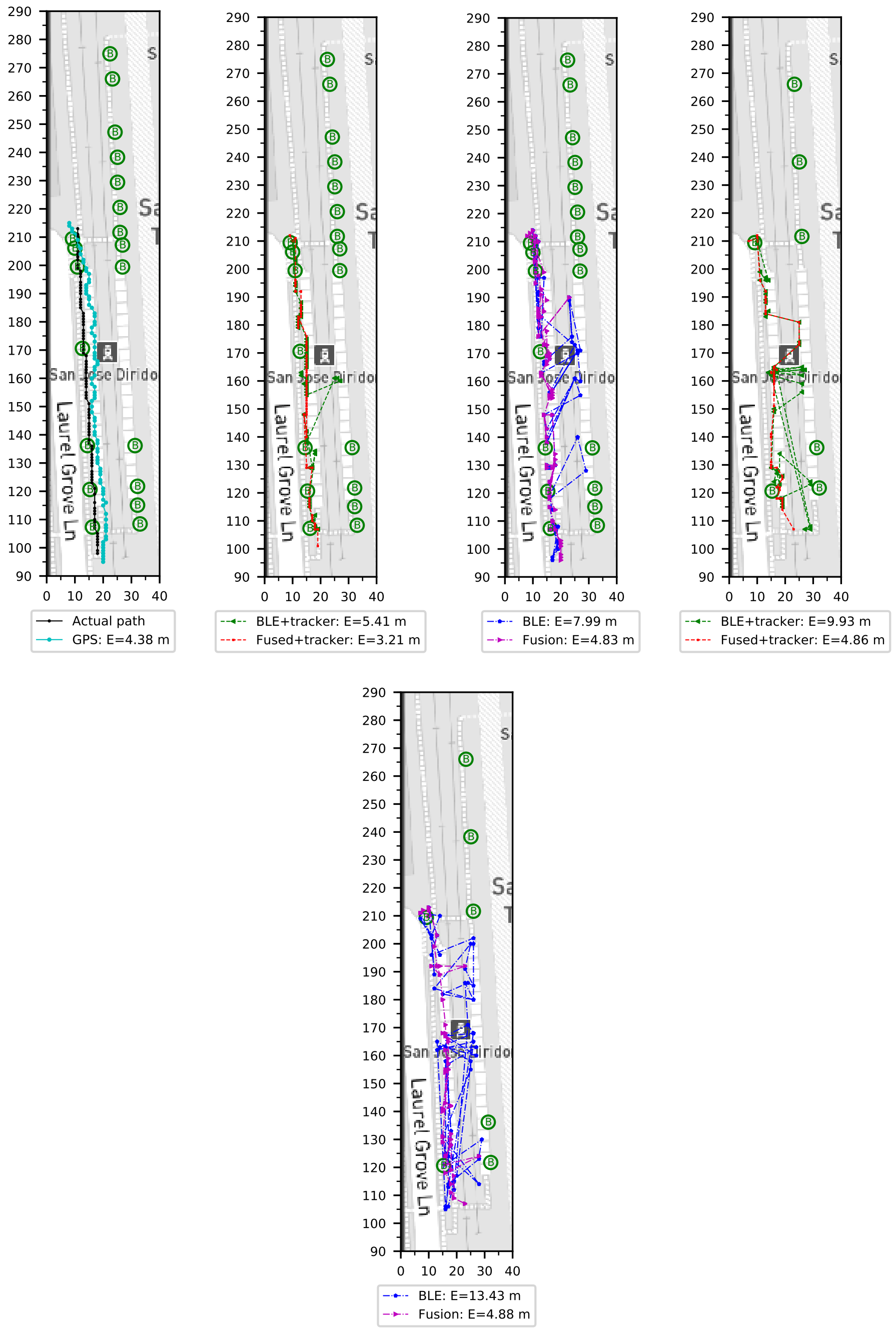


Figure 2.30: Experiments with a sample path (West platform) from Trial Set 2. See caption of Fig. 2.27.

GPS/beacon fusion results in substantially lower overall localization error than with either modality alone, especially when fewer beacons are used. Careful analysis shows that, in this case, fusion mostly contributes to reducing the extent of vertical (North-South) jumps from beacon-based localization. This may be the reason why no apparent benefit is observed from fusion with GPS with respect to using beacons alone in the “jump proportion" metric used for Trial Set 3. Vertical errors within the same segment do not constitute a jump, and are thus not penalized by this metric. In addition, fusion contributes to reducing East-West platform jumps when walking on the West platform.

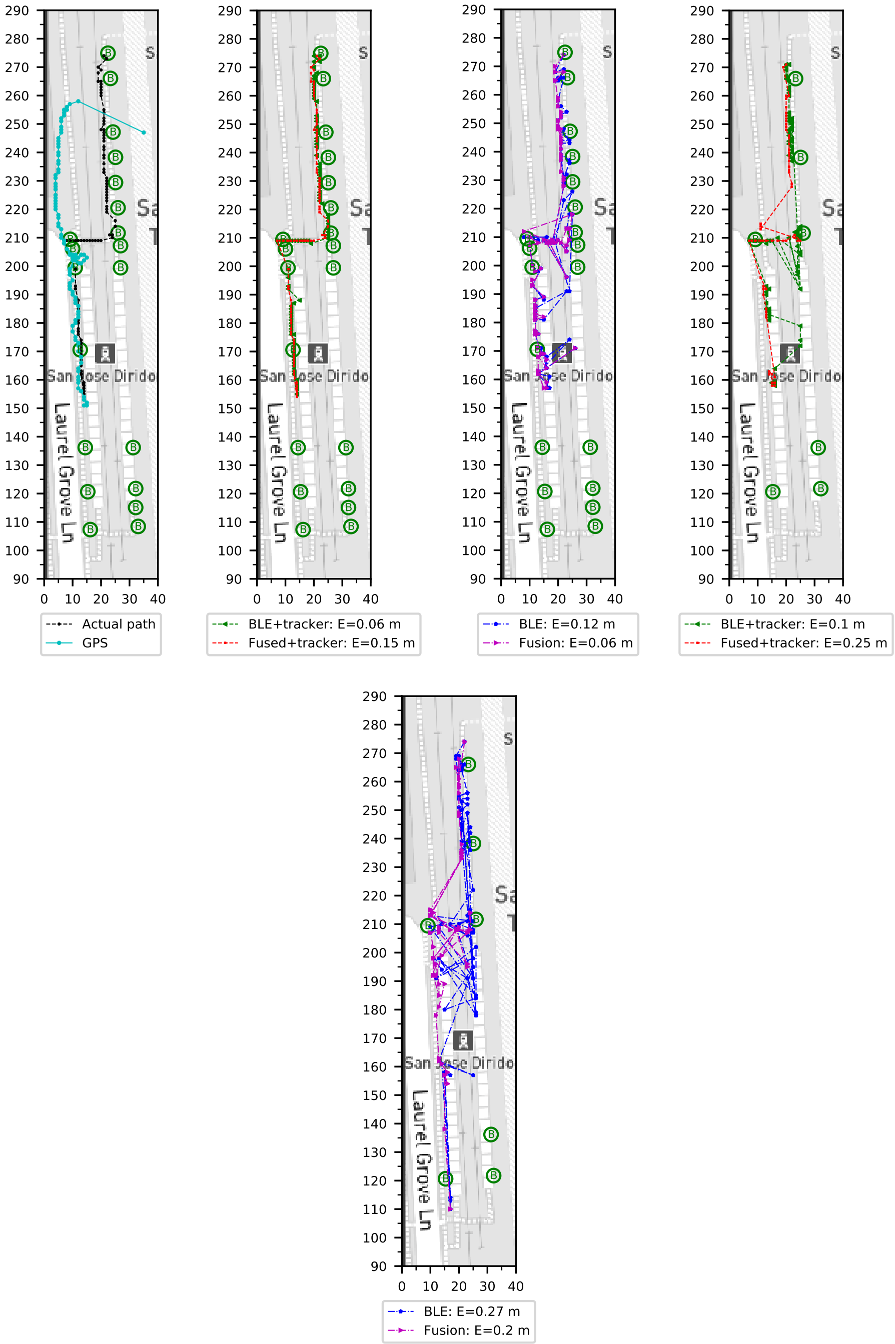


Figure 2.31: Experiments with a sample path from Trial Set 3. See caption of Fig. 2.27.

As expected, the general effect of the tracker is to “stabilize" and smooth the computed paths. Quantitatively, use of the tracker always reduces localization error, although in the case of Trial Set 2, the error reduction is less dramatic than that achieved by GPS/beacon fusion. Only for Trial Set 1 does use of the tracker on GPS/beacon fused data result in the best results. Some insight on why use of the tracker may not be as effective as expected in Trial Set 2 can be obtained by observation of Fig.2.29, which shows a case with me walking on the East platform. In this case, the tracker generates a piecewise smooth trajectory which sometimes places the user on the West platform, amplifying (rather than reducing) the localization error from the beacons. In this particular case, fusion with GPS may even worsen the situation, as seen in the case with 7 beacons.

2.13 Conclusion

I described a system for self-localization in an outdoor location (a light rail station) where GPS signal is available, but often unreliable. This situation is representative of many urban environments, where shading effects reduce the accuracy of GPS-based localization. A set of BLE beacons were installed, with the purpose to enhance self-localization through measurement of the RSSI from these beacons, using a mapping that was learned with a standard fingerprinting phase. Due to the open nature of the place, localization from BLE beacons alone is generally poor, especially when a low density subset of beacons is considered. Statistical fusion of data from GPS and beacons is shown to improve the accuracy in most situations, as does the use of a Bayes discrete filter tracker with a simple motion model.

Chapter 3

RouteMe2 at Palo Alto Station: User Experience Design

3.1 Introduction

In this chapter, I primarily focus on the experience of a visually impaired person who is using RouteMe2 mobile app to navigate in a complex transit hub. The main challenge is clear and precise communication between the localization system and a visually impaired user who does not have access to the contextual spatiotemporal information available solely by the gift of sight. Contextual refers to the visual information that is available in the surrounding of the user within a certain area such as a TL. The communication complexity becomes more challenging considering the inaccuracies and imperfections of the localization system in the open area, ramps, and tunnels.

For instance, in Palo Alto transit station, travelers have to navigate through underground tunnels, ramps, stairs, and outdoor areas to catch the train where there are possibilities for the user to fall off the track along the way. Designing a communication interface to communicate the spatiotemporal information along this kind of complex routes needs special care. A good user experience design shall be as simple as possible. It shall enable the visually impaired travelers to perceive their position, gain enough knowledge about their surroundings, and get notified of the upcoming actions to be taken. It also needs to consider the amount of the communicated information with respect to the cognitive load of the user [56]. Simply put, it must not overwhelm the user with too much information while she/he is busy with figuring out her/his way to the destination.

Fig. 3.1 shows a pedestrian route to Palo Alto Station. The route starts at Palo Alto transit center and ends at northbound train stop. A sighted traveler may be able to spot the correct bus slot by looking at the posted signs. However, a visually impaired traveler would likely need more directions to navigate the platform, find the place to get on the bus, and where and when to make turns for entering or exiting a ramp or an underground tunnel. All these adds complexity to the users' journey since they cannot rely on visual landmarks. Also, routes need to be defined at a much finer scale for visually impaired people. This is particularly the case in the open, when there are no readily available features that can be perceived by touch – such as a wall, which can be tracked using a long cane – to follow a route.

3.2 Chapter Organization

In section 3.3 I briefly review other accessible navigation systems built for people with visual impairments. In section 3.4 I present the user interface design thought process. Features and functionalities of the app is presented in section 3.5. Finally, I conclude the chapter in section 3.6.

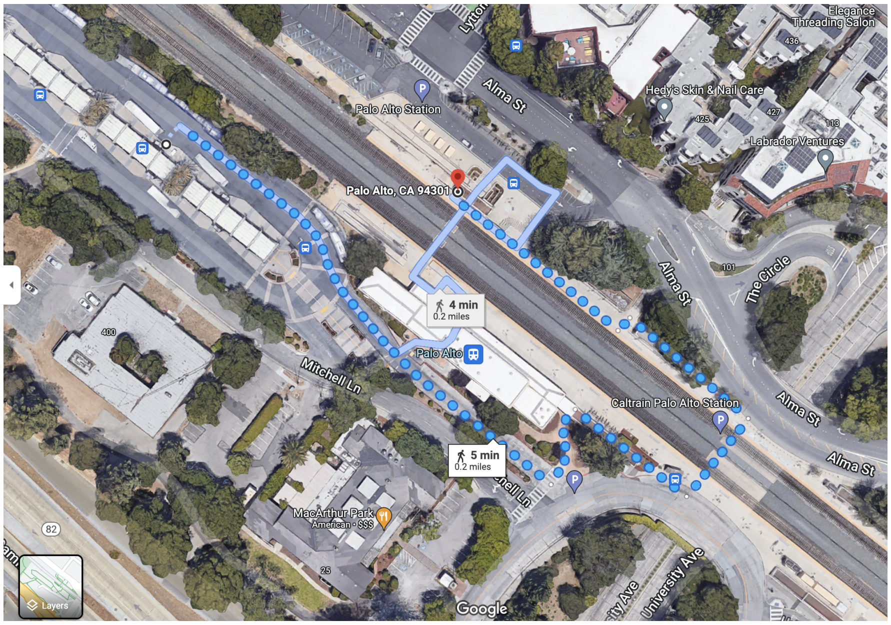


Figure 3.1: Detail from a route generated by Google Map. A pedestrian route, shown by blue dots, leads to a train stop from which the desired train departs (red destination icon). Finding where and when crossing the street, making correct turns to enter or exit a ramp or an underground tunnel are challenging for visually impaired.

3.3 Related Works

Here, I mention several existing navigation systems by concentrating on their user experience and app functionalities. The goal of this short review is to learn about the state-of-the-art and find out about the design limitations of these proposed systems and the areas of improvements. Table 3.1 summarizes these systems according to their user interface and functionalities and compares them with RouteMe2. A brief description of these system follows.

NavCog3 [41, 42] is a turn by turn navigation system for people with visual impairments for indoor environment. The app has plenty of adjustable settings that allow users to define their preferred route that may or may not include elevators, stairs, tactile pavement, etc.

Sightless Helper [57] uses footstep counting and GPS for indoor and outdoor navigation. It can detect unsafe areas to ensure safe navigation. It allows users to define unsafe areas by finding them on Google Maps. They evaluated the usability of the system using System Usability Scale (SUS) score proposed in [58] and reported 72.2% as their score. It provides general guidance toward point of interests defined by users. It lacks detailed contextual information and proper instructions from user experience perspective. For example, it does not provide orientation and heading direction which is essential for visually impaired users.

BlindSquare [59] is a commercial accessible GPS-app developed for the blind, deafblind and partially sighted. It pairs with third-party navigation systems such as Google Maps, Apple Maps, and Moovit for navgation. It uses Foursquare database to find nearby point of interests. It is more of an exploratory app. It has plenty of adjustable settings. However, it does not provide turn by turn navigation required for people with visual impairments.

Blindways [60] guides users within a cane's distance of an outdoor bus stop sign using permanent landmark clues contributed by volunteers. Dependency of the model to the volunteers contributions may decrease the usability of the app in every stop.

	[41, 42]	**[57]**	**[59]**	**[60]**	**[61]**
Name	NavCog3	Sightless Helper	Blindsquare	Blindways	RouteMe2
Indoor Outdoor	*Indoor*	Both	Outdoor	Outdoor	Outdoor
Localization	*BLE*	*GPS*, footstep counting	*GPS*	*GPS*, *Clues*	*GPS*, BLE / IMU
Heading direction detection	*No*	*No*	*No*	*No*	*Yes*
Turn by turn instruction	*Yes*	*No*	*No*	*No*	*Yes*
Safe Traversable Area	*No*	*Yes*	*No*	*No*	*Yes*
Supports public transit	*No*	*No*	*No*	voluntarily landmarked bus stops only	*Yes*
Voice output	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*
Vibration feedback	*Yes*	*Yes*	*Yes*	*No*	*Yes*
Interaction style	Conversational	Instruction	Instruction	Instruction	Instruction
POI and Landmark announcement	*Yes*	*Yes*	*Yes*	*Yes*	*Yes*

Table 3.1: Summarizes accessible navigation apps functionalities and compares them with RouteMe2.

My investigation on these apps resulted in the following conclusions. 1. Main page of most successful systems shows a map and routing, 2. voice output and vibration feedback are common features that users expect, 3. None of the current systems are useful for complex transit hubs, 4. There is lack of step by step routing or turn by turn navigation app for outdoor complex transit centers where GPS is not accurate, 5. Simple graphical user interface is lacking.

3.4 User Interface Design

The first and foremost step toward designing a user interface that leads to better user experience for visually impaired users is to acknowledge the fact that they cannot see. Despite the simplicity of this fact, it is surprisingly a challenging task for a sighted designer to put herself in the shoes of a visually impaired person and truly feels what her/his experience is while using the app. Also, there are different scenarios some of which may misleadingly seem trivial but each have its own complications. For instance, when the visually impaired user is off track, is near stairs, or has to take consecutive turns. On top of that, there must be a medium for the designing team to communicate the user experience with one another and imagine the experience of the visually impaired user as they collaborate on the designs. In attempt to meet the preceding demands, I resorted to a some storytelling techniques by developing personas and scenarios which resulted in a set of storyboards. These storyboards provided us a medium to communicate among our team in the lab and helped us to imagine the situation where the visually impaired user may encounter during their journey in the transit hub.

A persona is a description of a fictitious user, based on data from user research [62]. Scenarios are the stories that we tell while we have the persona of our users in mind. Storyboards provide a common visual 'language' that enables people from different backgrounds to communicate on aspects of design [63].

I define a persona called Julia who is 53 years old and she is blind. She travels from *Palo Alto transit center to Northbound Palo Alto station* once a week. Figures 3.2 - 3.11 show the scenes of a storyboard about her journey from Palo Alto transit center to Northbound Palo Alto station I refer to it as *PA:TC-NB* which is listed below. It is worth noting that the TL numbers in the route descriptions in those figures are replaced with actual TL names. For example, *underground south tunnel* or *south cross walk*.

- When the user gets lost,

- When the user wants to know where she/ he is?,
- When the user wants to learn what is around her/ him?,
- When the user wants to learn direction toward desired distal landmark,
- When the user wants to know her/ his current location and orientation,
- When the user wants to know about upcoming turns,
- When the user wants to know about the distance left to the destination,
- When the user wants to know about the details of the route either before starting the route or somewhere in the middle of the route.
- When the user is walking in a wrong direction and not toward the next tile,
- When the user is not in traversable area and off route.

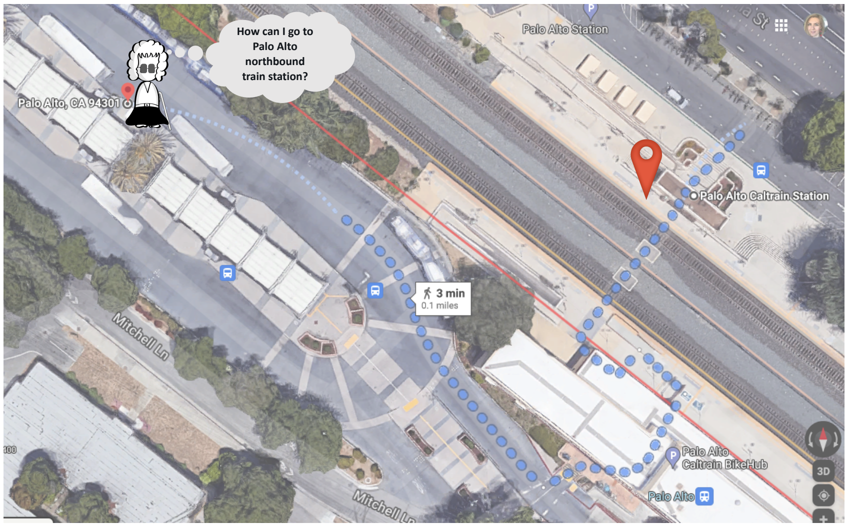

How can I go to Palo Alto northbound train station?
Palo Alto, CA 94301
Palo Alto Station
Palo Alto Caltrain Station
3 min
0.1 miles
Mitchell Ln
Mitchell Ln
Palo Alto Caltrain BikeHub
Palo Alto
3D

Figure 3.2: *start* scene of *PA:TC-NB* storyboard is demonstrated. Julia (the hypothetical user) is at a bus stop at Palo Alto transit center and does not know how to get to the northbound train station.

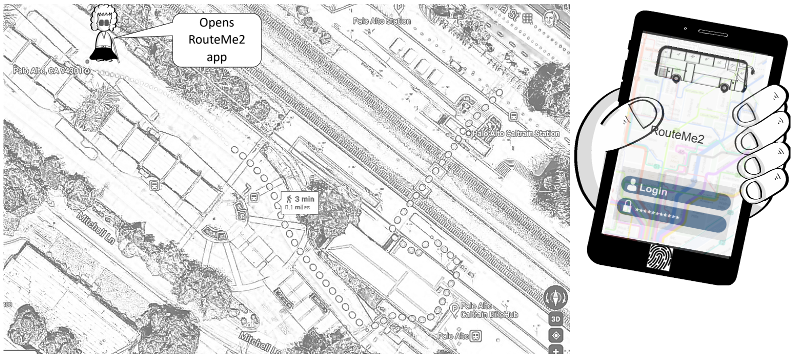


Figure 3.3: *open RouteMe2* scene of *PA:TC-NB* storyboard is demonstrated. Julia (the hypothetical user) opens the app and logs in with her credentials.

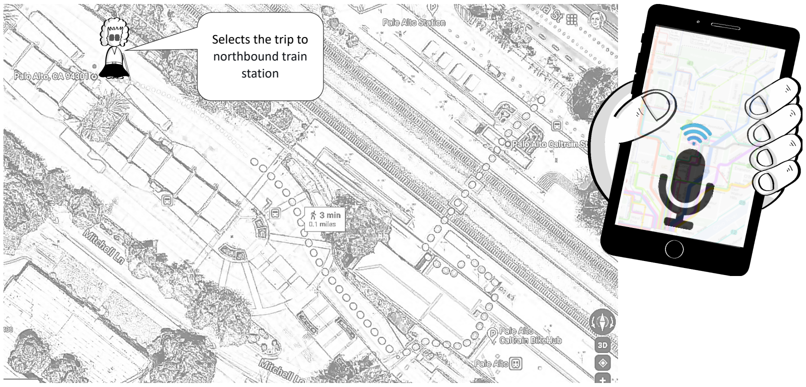
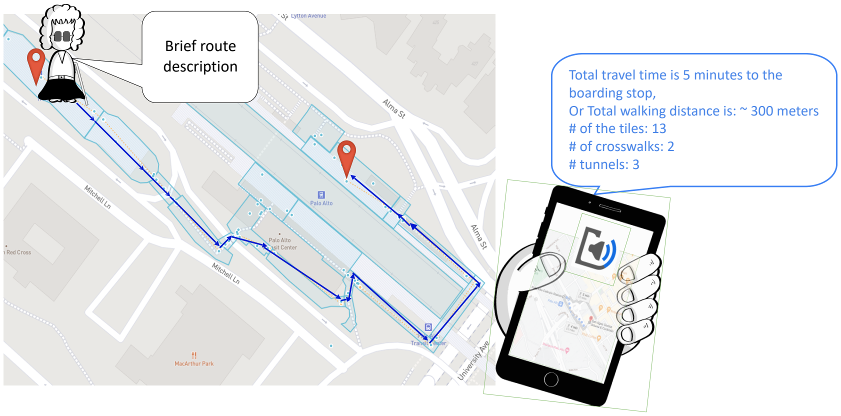
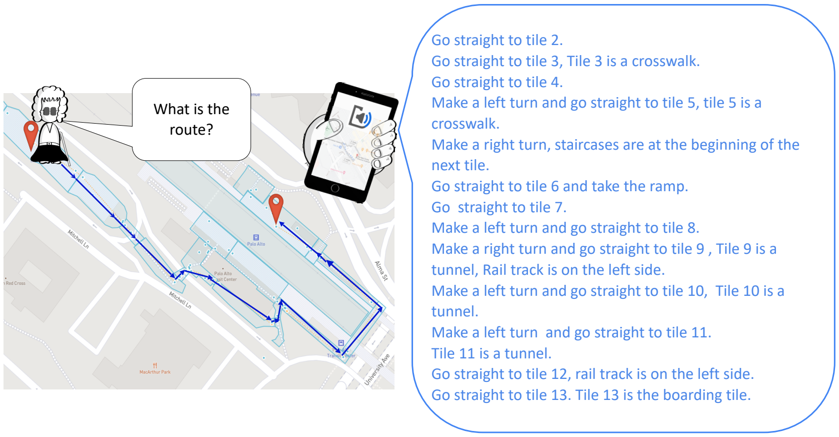


Figure 3.4: *select a trip* scene of *PA:TC-NB* storyboard is demonstrated. Julia (the hypothetical user) selects her trip which was defined by herself or her care giver in RouteMe2 web application



Total travel time is 5 minutes to the boarding stop,
Or Total walking distance is: ~ 300 meters
# of the tiles: 13
# of crosswalks: 2
# tunnels: 3

Figure 3.5: *brief route description* scene of *PA:TC-NB* storyboard is demonstrated. Julia (the hypothetical user) wants to hear a brief route description before staring the route.



Go straight to tile 2.
Go straight to tile 3, Tile 3 is a crosswalk.
Go straight to tile 4.
Make a left turn and go straight to tile 5, tile 5 is a crosswalk.
Make a right turn, staircases are at the beginning of the next tile.
Go straight to tile 6 and take the ramp.
Go straight to tile 7.
Make a left turn and go straight to tile 8.
Make a right turn and go straight to tile 9 , Tile 9 is a tunnel, Rail track is on the left side.
Make a left turn and go straight to tile 10, Tile 10 is a tunnel.
Make a left turn and go straight to tile 11.
Tile 11 is a tunnel.
Go straight to tile 12, rail track is on the left side.
Go straight to tile 13. Tile 13 is the boarding tile.

Figure 3.6: *complete route description* scene of *PA:TC-NB* storyboard is demonstrated. Julia (the hypothetical user) wants to hear more about the details of the route such as upcoming turns where she expects to enter to a tunnel, etc.

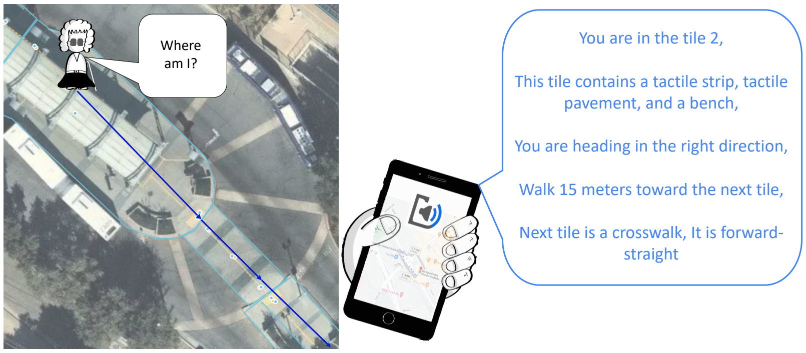

Where am I?
You are in the tile 2,
This tile contains a tactile strip, tactile pavement, and a bench,
You are heading in the right direction,
Walk 15 meters toward the next tile,
Next tile is a crosswalk, It is forward-straight

Figure 3.7: *where am I* scene of *PA:TC-NB* storyboard is demonstrated. Julia (the hypothetical user) is confused and wants to hear where she is.

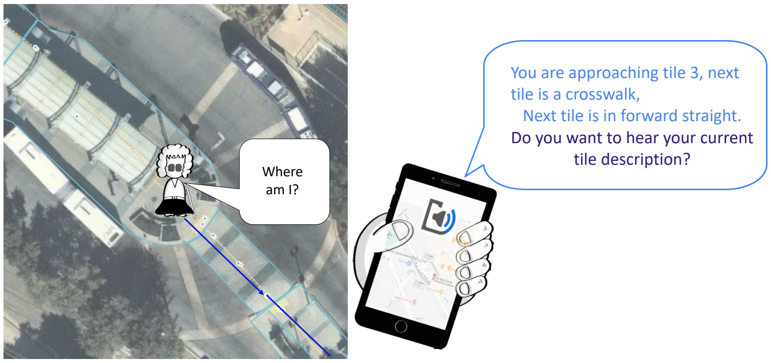

Where am I?
You are approaching tile 3, next tile is a crosswalk,
Next tile is in forward straight.
Do you want to hear your current tile description?

Figure 3.8: *approaching crosswalk* scene of *PA:TC-NB* storyboard is demonstrated. Julia (the hypothetical user) is confused and wants to hear where she is. She is about to cross the crosswalk.

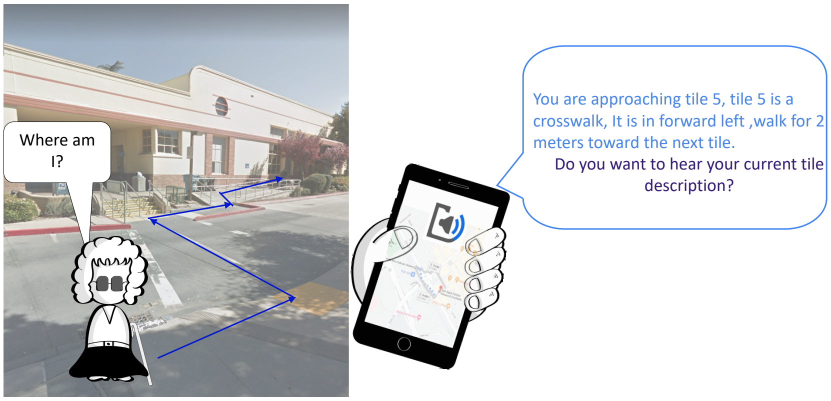

Where am I?
You are approaching tile 5, tile 5 is a crosswalk, It is in forward left ,walk for 2 meters toward the next tile.
Do you want to hear your current tile description?

Figure 3.9: *where am I* scene of *PA:TC-NB* storyboard is demonstrated. Julia (the hypothetical user) is confused and wants to hear where she is. She is about to turn left to cross the crosswalk.

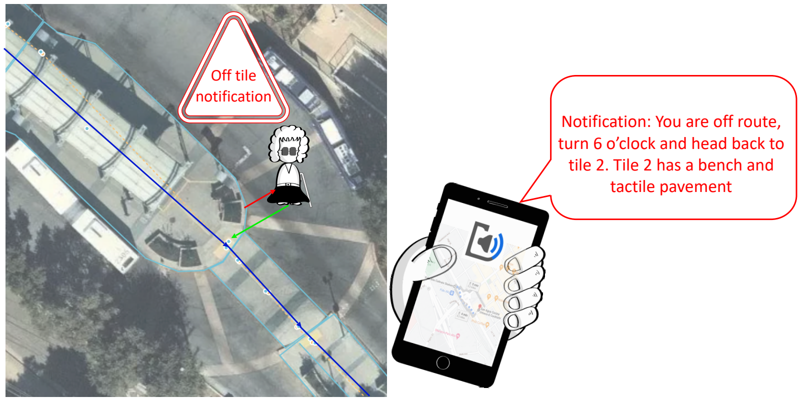
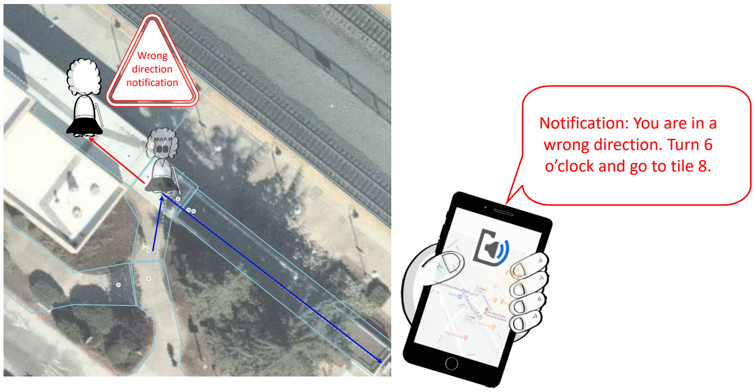

Off tile notification
Notification: You are off route, turn 6 o'clock and head back to tile 2. Tile 2 has a bench and tactile pavement

Figure 3.10: *off route* scene of *PA:TC-NB* storyboard is demonstrated. Julia (the hypothetical user) is off route and the app notifies her. The app provides instructions to return to the route.



Notification: You are in a wrong direction. Turn 6 o'clock and go to tile 8.

Figure 3.11: *wrong direction* scene of *PA:TC-NB* storyboard is demonstrated. Julia (the hypothetical user) walks in an opposite direction. The app identifies it and notifies her. The app provides direction instructions to correct her heading direction.

The design process included building storyboards, wireframing, and mock-ups. After collaboratively working on the scenes of a storyboard, wireframing was the second step toward the app user interface design. The design process is an interactive and iterative process in which I circle back and update the scenes and the design based on the feedback received from our team, identifying new user's needs, and new features.

3.5 RouteMe2 app Functionality

Based on the storyboard presented in section 3.4 and our findings from our previous focus group study [3], I prioritized four main functionalities that cover some of the challenges people with visual impairments might have in outdoor places such as a complex transit hub. The notion of tile or TL may be used interchangeably. Those functionalities are as follows:

- Brief route description,
- Route description,
- Where am I?,
- What is around me?

These functionalities are state dependent and they provide different outcomes based on the user's state. I defer the description of these functionalities to after I describe the states below since these functionalities are state dependent. The system state diagram 3.12 is created to consider all the states that our user-centric app should handle for different functionalities.

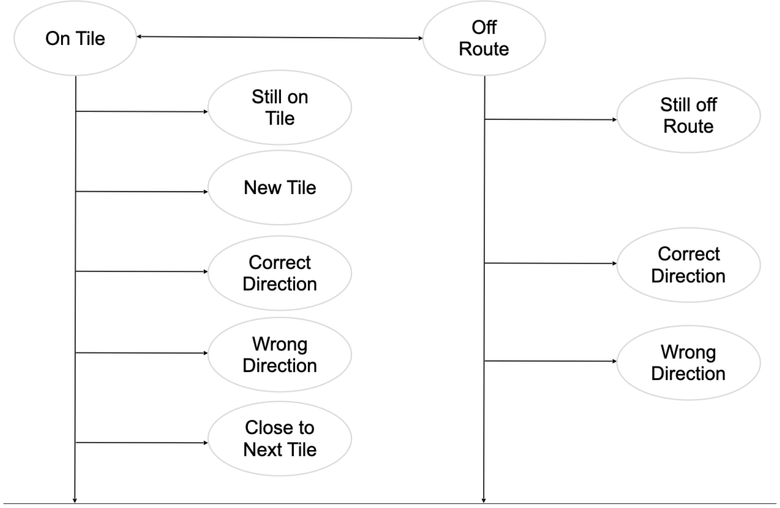


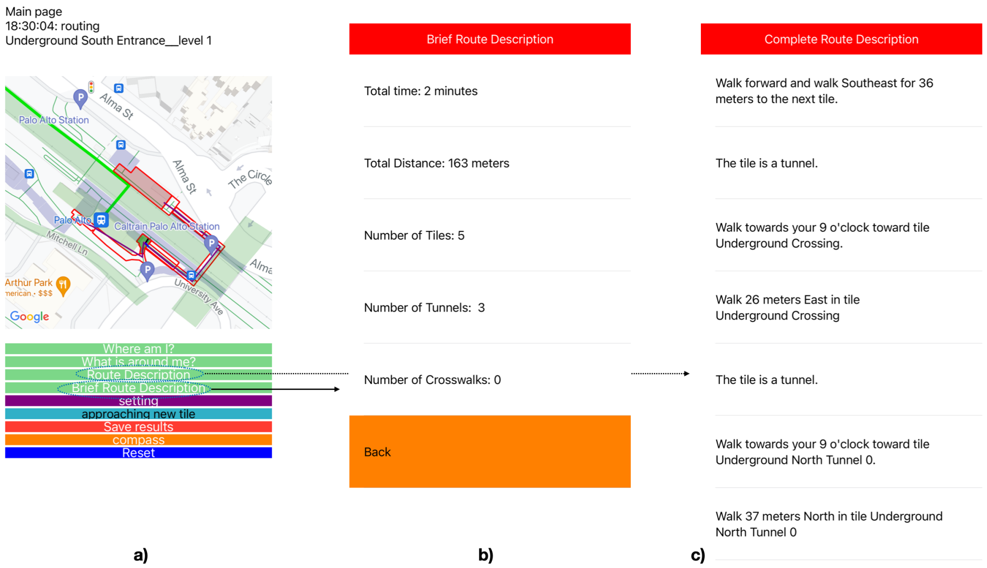
Figure 3.12: System states diagram that shows the change of the states based on user location, heading direction, and route information.

The user can be in either of *on Tile* or *off Route* state. The app detects the location and heading direction of the user. It checks them against the routing information that includes list of TLs on the route. List of TLs are generated in sequential order from user's location toward the destination. The app detects whether the user is on any TL in a route or is completely off route and needs instructions to get back to the route.

on Tile state itself has its own sub states. Still on tile state happens when the user is on route and in correct tile toward next tile. It is worth mentioning that every time the app detects the tile that the user is within (current active tile), the system finds the next and previous tiles to provide correct instructions. When user approaches the next tile in the route, the state of the system changes to *close to next tile*. Here the app vibrates once to let users get updates about their current location along the route. Then, user enters the next tile and app vibrates twice and changes its state to *new tile* – i.e. next tile. The app constantly computes the heading direction of the users and compares it with the correct heading direction toward next tile. *Correct direction* state is for when the user walks in the correct direction toward next tile and *wrong direction* state happens when a user walks in a correct tile but wrong direction. When this situation happens, the app state changes to *wrong direction* state. The app looks for a new route if available to update the route based on current location and orientation of the user. If not, it generates new direction instructions and asks the user to change her/ his heading direction toward next correct tile.

As mentioned earlier *off route* state happens when user is completely off route and not detected in any tile. The user might have been in a correct tile and walked out of the tile and not being on the route anymore. In this situation the heading direction is checked. The user either can be in a correct direction toward closest tile in the route or completely in a wrong direction. The state of the system changes to *off route - correct direction* in the former situation and *off route - wrong direction* in later case. In both case the app generates navigation instructions to help the user to successfully get back to the previous tile or closest tile in the route. In *off route - wrong direction* situation the app vibrations four times to alert the user. The system stays in *still off route* state until the traveler enters the correct tile in the route.

Now back to the functionalities of the app. *Brief route description* provides important information that user may wants to know either before starting the route or along the route. Information such as number of tiles (TLs), number of tunnels, number of crosswalks, duration of the route, and total time takes from the staring point to the destination (Fig. 3.13, b)). *Route description* lists step by step routing guidance from the current location of the user to the destination. *Route description* provides walking direction toward next tile, walking distance toward next tile, heads up about important tiles such as a cross walk, stair cases, ramps, and tunnels (Fig. 3.13, c)). *Where am I?* feature can be used anytime including being on tile or on route or being off route. It returns the current tile that the user is in and a direction instruction toward the next tile when system state is on tile. On off route state it generates walking and direction instructions toward correct tile. It also provides whether next tile is an important tile such as a tunnel, cross walk, or staircase (Fig. 3.14, b)). *What is around me?* feature returns list of important points of interest and landmarks based on the current location of the user along with routing information such as distance and direction toward them (Fig. 3.14, c)).



Main page
18:30:04: routing
Underground South Entrance__level 1

Brief Route Description

Total time: 2 minutes

Total Distance: 163 meters

Number of Tiles: 5

Number of Tunnels: 3

Number of Crosswalks: 0

Back

Complete Route Description

Walk forward and walk Southeast for 36 meters to the next tile.

The tile is a tunnel.

Walk towards your 9 o'clock toward tile Underground Crossing.

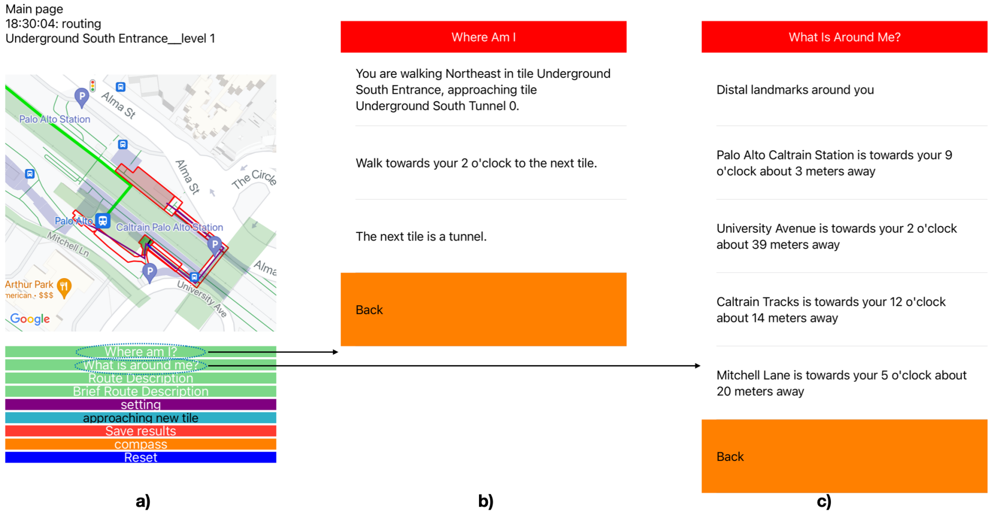
Walk 26 meters East in tile Underground Crossing

The tile is a tunnel.

Walk towards your 9 o'clock toward tile Underground North Tunnel 0.

Walk 37 meters North in tile Underground North Tunnel 0

Figure 3.13: User interface of the RouteMe2 app is demonstrated. a) user selects either *brief route description* or *route description*. b) App shows the brief route information. c) App shows the route description.



Main page
18:30:04: routing
Underground South Entrance__level 1

Where Am I

You are walking Northeast in tile Underground South Entrance, approaching tile Underground South Tunnel 0.

Walk towards your 2 o'clock to the next tile.

The next tile is a tunnel.

Back

What Is Around Me?

Distal landmarks around you

Palo Alto Caltrain Station is towards your 9 o'clock about 3 meters away

University Avenue is towards your 2 o'clock about 39 meters away

Caltrain Tracks is towards your 12 o'clock about 14 meters away

Mitchell Lane is towards your 5 o'clock about 20 meters away

Back

Figure 3.14: User interface of the RouteMe2 app is demonstrated. a) user selects either *where am I?* or *what is around me?*. b) App shows the current location information along with direction toward next tile. c) App shows the list of landmarks ordered by their vicinity to the current location of the user. It also provides direction information toward them.

3.6 Conclusion

In this chapter I presented my design philosophy and the thought process behind the user experience and interface designs. My goal was to provide an effective medium for communicating the spatiotemporal information of the surrounding environment, the state of user in that environment, and navigation and guidance to the visually impaired user. The user interface is designed having the simplicity of the graphical interface and the requirements that we identified in our focus group study [3] in mind.

I strived to put the user at the center of my designs by creating storyboards to cover important real world scenarios in different situations that the user may encounter during an actual journey from a source to a destination in an actual complex transit hub. Then, I deduced the required functionalities to cover those scenarios depending on the user's state. Each step of the design required considerable amount of brainstorming and back and forth with the team.

Due to time and resource constraints, we were not able to involve actual users in the design process which would be ideal if we could have afforded to do so. For the same reason, I was not able to assess the success of the designs in meeting the mentioned objectives by means of user study, usability testing, focus group, or other feedbback methods.

Chapter 4

Interpersonal Proximity Detection and In-Vehicle Localization Using BLE

4.1 Introduction

The COVID-19 pandemic has affected virtually all enterprises in the private and public sector. In particular, public transit has suffered disproportionally from loss of ridership [64, 65]. As a consequences of shelter-in-place ordinances, and with remote working becoming accepted and even encouraged in many lines of business, the commuting needs of many habitual bus or train riders have radically reduced. In order to enforce social distancing, transit operators have been forced to dramatically reduce the capacity of vehicles. Many potential riders are choosing not to use public transit for fear of contagion, even though there is scant evidence that, with appropriate precautions in place, transit poses serious risks of coronavirus outbreaks [66].

Yet, many people (in particular, essential workers who cannot afford private transportation) are still riding busses and trains. And once the pandemic will be under control, it is expected that ridership will increase again. Indeed, public transit has a critical role for sustainable, affordable, and accessible mobility [67]. Even in the era of autonomous vehicles, mass transit will be necessary to manage traffic congestion [68]. In the words of Jeff Tumlin, Director of Transportation at San Francisco Municipal Transportation Agency (SFMTA): “Transit remains the most energy and space efficient way to move large numbers of people over long distances in and around cities” [69].

However, transit riders in the post-pandemic world will have increased expectations. Prophylactic measures, such as maintaining social distancing and avoiding touching surface in common places, are likely to remain on the mind of travelers. Agencies will need to put policies and infrastructure in place that make riders feel safe and comfortable while using public transit [70, 71].

In this chapter I addresses three interconnected services contributing to a safe travel experience: *interpersonal proximity detection*, *effortless ticketing*, and *crowdedness monitoring.*

4.2 Interpersonal Proximity Detection

Interpersonal proximity detection techniques, once confined to applications such as crowd monitoring [72] and social interaction analysis [73], have received substantial recent attention due to their potential for COVID-19 contagion tracing. Contact tracing may help understand the genesis of a local outbreak of the disease, and could be used to warn subscribers about a potential contagion event due to proximity with an infected person [74].

The most common approach for interpersonal proximity detection relies on measurement of the received signal (RSSI) from a radio source, such as a Wi-Fi transmitter [73] or a Bluetooth Low Energy (BLE) beacon [75–79]. Since all modern smartphones contain a BLE transceiver, this approach has enabled widespread adoption without the need for expensive external infrastructure (e.g. cameras equipped with embedded computers for visual people tracking.) The power of the received signal decreases quadratically with the distance D to the source, and thus the received strength could, in principle, be used to estimate D when the emission power is known. For example, the Exposure Notification (EN) API produced by Google and Apple [80] uses this mechanism to support contact tracing.

A different approach to proximity detection is based on the *disparity* of a measured signal, typically an electromagnetic field generated by a transmitted (e.g., a Wi-Fi [81–83] or BLE [84, 85]), although magnetic field measurements have also been considered [86, 87]. When two identical receivers are placed in the same or similar location, measurements are expected to be similar (i.e., their disparity, as defined, for example, by the magnitude of their difference, is expected to be small).

I present an experimental comparative analysis of mechanisms that use measurement disparity (of the RSSI from fixed BLE beacons) for proximity detection, viz-a-viz the direct measurement of RSSI from another nearby smartphone. In particular, I address the specific problem of detecting the presence of another individual within distance thresholds of 1 meter and of 2 meters, since these are the interpersonal distances usually considered when establishing the risk of contagion [4, 5]. Unlike other work on proximity detection, where traces of moving individuals are analyzed to identify possible overlaps, I consider the case in which two individuals are standing or sitting at certain distance from each other for a period of time. This is representative of typical contagion scenarios, such as sitting at nearby tables at a restaurant, or in nearby seats in a bus vehicle. I focus solely on RSSI data here. Although data from other sensors (e.g. inertial [84, 85]) can be leveraged to reduce false positives, I believe that it is important to precisely assess the contribution of each modality.

The principal contributions of my work flows:

1. I collected representative data sets from two different environments: a living room, instrumented with three BLE beacons, and a campus shuttle bus with four BLE beacons. Within each environment, multiple data collection sessions separated by long periods of time were conducted in order to assess repeatability.

2. I present an in-depth statistical analysis of the data collected, and of its ability to discriminate interpersonal distance using a threshold of 1 meter and of 2 meters.

3. I compare the system performance using different features (including RSSI from another phone, individual and average RSSI disparities from multiple BLE beacons), as well as of a simple additive combination of RSSI received from another phone and of mean RSSI disparity.

4.2.1 Related Work

Two approaches for contact tracing systems have been proposed in literature. 1) Network-based sensing approach that requires no client-side involvement and uses WiFi infrastructure to passively monitor the flow and mobility of people in a region equipped with WiFi Access Points (AP). WiFi networks log the connections of mobile devices to APs and infers crowd movement patterns across a region along with occupancy levels in different buildings by analyzing the number of smartphones connecting to each AP [72, 81, 88, 89]. However, there are some issues associated with network-based systems. First, one generally has no control on the actual density of AP placements that results in lack of coverage for some areas of interest. Second, the long range coverage of WiFi APs are not helpful for contact tracing purposes when higher positional resolution is required [82]. Lohan *et al.* [90] reported that the short operating range technologies such as BLE typically provides better performance than WiFi positioning in terms of the estimated distance/ranging error. 2) Client-based sensing that requires users to install an app and uses smartphone sensors and BLE data to perform sensing measurements. BLE technology is well justified to be used in contact tracing systems due to its availability on most smartphones, low cost, and energy efficient [91]. In the context of social interaction measurement previous studies [75, 77, 84, 92] used Bluetooth RSSI either from smartphones or wearable sensors (such as smartwatches or coin beacons) by mapping RSSI to distance via *propagation* model. [76] and [91] employed wearable devices to simulate smartphone BLE measurements in order to circumvent the iOS limitations in BLE scanning when the app is in background mode.

In principle, it could be possible to use power decay models (a.k.a. propagation models) [24, 25] to estimate the interpersonal distance from the measured RSSI received from the other device. However, in practice, this is extremely challenging [9] due to issues such as multipath fading – an effect of signal reflection from nearby surfaces, time dependent signal power variations, and diversity of smartphones/environment in the contact tracing context. Power decay models require calibration for each environment to identify a path loss coefficient for each device. This is not practical in contact tracing systems. [75] studied face-to-face proximity estimation using power decay models based on Bluetooth in smartphones. They showed that even in ideal situations (indoors, same antenna orientation), the RSSI signals measured within 1.5 meters to 3.5 meters are practically indistinguishable.

[82, 84, 93, 94] combined different modalities such as Ultrasound, Radio frequency identification (RFID), Quick Response (QR) codes, short range Wifi APs, and external BLE beacons to improve the accuracy of the inter-personal distance detection. Along the same lines, Shankar *et al.* and Trivedi *et al.* [85,88] leveraged BLE beacons, WiFi, and Ultrasound co-location technologies, but in the context of social interaction detection.

To the best of my knowledge no one has studied the use of external beacons along with smartphone BLE RSSI data in the context of interpersonal distance estimation for contact tracing purposes in a bus. Leith *et al.* [95] ran experiments on a commuter bus and analysed the phone BLE signals provide by EN API. They observed that increasing the exposure duration improves the accuracy of their model at the cost of reducing the time resolution of the distance detection system. They hypothesised that the sample rate provided by EN API is not sufficient for a practical interpersonal distance detection in the bus where the signal propagation is under the influence of disturbances that arise from metal-rich environments. Hence, further studying the problem in this challenging environment by considering higher phone BLE sample rates as well as combining that with other modalities such as external BLE beacons will get us closer to a practical solution.

4.2.2 Interpersonal Proximity Detection Techniques

In this experiment, I considered two approaches to interpersonal proximity detection. *Phone RSSI* uses the strength of the BLE signal received from another persons' smartphone. *RSSI disparity* compares the signal strength received at the same time and from the same BLE beacon by the smartphone carried by two individuals.

Phone RSSI

In principle, one could use power decay models to measure the distance to a transmitter from the measured RSSI [24, 25]. For example, a distance–dependent threshold could be devised as a function of the transmitter's characteristics. In practice multiple factors (including signal absorption from human bodies, reflection from walls and other obstacles and the orientation of the receiving antenna) cause substantial deviations from the model. This especially the case for indoor environments [78].

RSSI Disparity

BLE beacons are often installed in public spaces such as airports and shopping malls, for applications such as mobile advertising or to enable self-localization [34, 41, 42]. This existing infrastructure can be leveraged for proximity detection. One approach could be to use the RSSI from multiple beacons to localize the user via fingerprinting techniques [26–29, 32], then using this data to verify whether two users where in nearby locations at the same time. However, fingerprinting information is not normally available, and accuracy of localization may be poor. A simpler approach can be used based on the notion that if two persons are co-located, the power received from a beacon should be similar for both users [81–83]. Thus, by comparing the RSSI from multiple beacons for two (or more) users, one could devise a *disparity index* that measures the pairwise difference in received RSSI. This disparity index could be associated with the likelihood of these individuals being within a certain distance to each other.

It is important to note, though, that a small value of disparity does not necessarily mean that the individuals are co-located. Ambiguity may arise when the same signal power is measured in different locations. In order to illustrate how this can happen, consider the ideal case of isotropic signal loss, whereby the measured signal power is only a function of the distance to the beacon. Two different locations may result in the same measured power from a beacon when they are at the same distance to the beacon. In the case of a single beacon, all locations within the same circle around the beacon are ambiguous (Fig.4.1 (a)). Ambiguity can be reduced or eliminated by using more beacons. In the case of two beacons, for a given the location of one person, there is exactly one other location from which the same signal strength is received (i.e., that is whose distance to each beacon is the same as for the first person; Fig.4.1 (b)). Using more beacons, the ambiguity is resolved (i.e., two persons in different locations will receive a different signal strength from at least one beacon; Fig.4.1 (c)), unless the beacons' locations are collinear (Fig.4.1 (d)). In practice, the relationship between distance to a beacon and received signal strength is affected by the same factors mentioned in Sec. 4.2.2, which may contributes to the inherent ambiguity of this method for proximity detection.

In the experiments, I use as disparity index the absolute value of the difference $d_j = |RSSI_j(1) - RSSI_j(2)|$ of the received strength from the j-the beacon by the two smartphones. In the case of signal received from multiple beacons, I simply consider a one-dimensional feature formed by the average value the individual disparity (akin to the Manhattan distance considered in [81]). This is a reasonable choice, considering that all individual disparity values (and thus their mean) are expected to be small at small interpersonal distances. Other choices (e.g., taking the max value of the disparities) did not give good results in the preliminary tests.

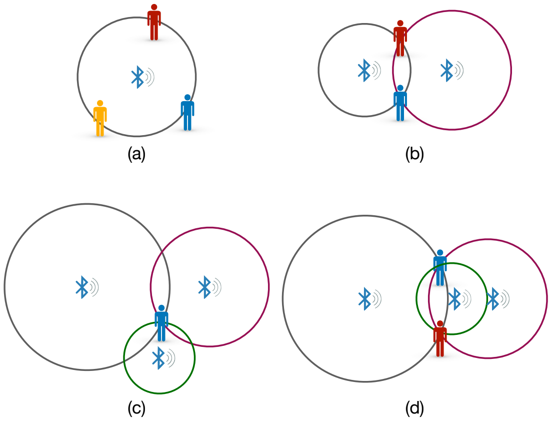


Figure 4.1: Examples of ambiguous zero RSSI disparity situations using one beacon (a), two beacons (b), and three or more collinear beacons (d). Ambiguity can be avoided by using three or more non-collinear beacons (c). These examples assume isotropic signal loss and uniform emission power.

4.2.3 Experiments

A Toy Case: Instrumented Living Room

In order to evaluate the considered proximity detection techniques in a simple, controlled scenario, I instrumented a living room with three BLE beacons (Kontakt Tough Beacon TB15-1) configured as iBeacons and set to the power level 2 (RSSI of -81 dBm at 1 meter) and advertisement interval of 350 ms. The beacons were placed at a height of 2.5 meters, at the locations shown in Fig. 4.2. One iPhone 7 and one iPhone 8 were used in the study. An app was installed in each phone, designed to record time stamped RSSI data from the other phone's BLE beacons, as well as from the external BLE beacons. Phone RSSI and BLE beacons reading rate were 40 and 3 samples per second respectively. I scanned Phone RSSI and BLE beacons measurements using Core Bluetooth and Core Location frameworks respectively. Data was collected while two experimenters, carrying one iPhone each, stood at different locations as shown in Fig.4.2, with interpersonal distance of 1, 2, 3, and 4 meters. For each location pair, the experimenters first collected data for two minutes while holding the smartphones in their hand; then they placed their iPhones in their front pants pocket and collected data for two more minutes. The experimenters faced each other at all times. This data collection was repeated three times: in May of 2020 (*Set 1*), in December of 2020 (*Set 2*), and in April of 2021 (*Set 3*). Data sets from each location pair and each phone placement were compiled together into a single sequence (*Combined* set).

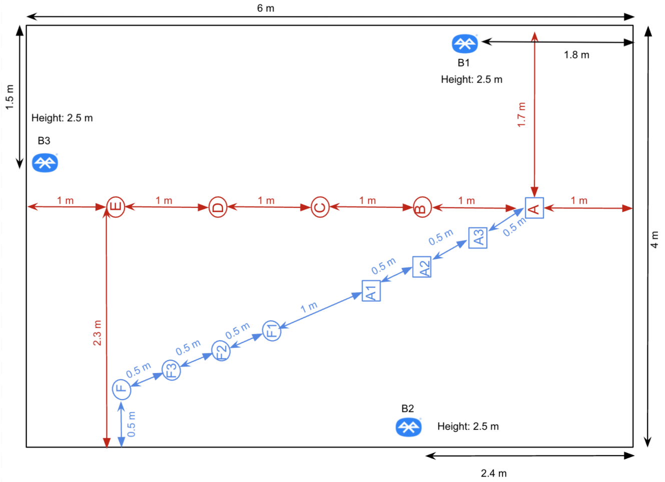


Figure 4.2: Home data collection layout: Set 1 data collection at (A1, F1), (A2, F2), (A3, F3), (A, F) location pairs at 1, 2, 3, and 4 meters interpersonal distance respectively. Set 2 and Set 3 data collection at (A, B), (A, C), (A, D), (A, E) location pairs at 1, 2, 3, and 4 meters interpersonal distance respectively. For Set 2 and 3, one experimenter stood still in position A, while the second experimenter moved in turn on the other locations.

The data collected at each location pair and for each phone placement was preprocessed as following. Missing RSSI measurements were replaced with a small value (-100 dBm). The resulting sequence was run through median filter of length 15, in order to reduce the variance of noise.

In order to analyze the statistical characteristics of the various indicators considered, I plotted the probability density function (pdf) of the associated measurements conditioned on the interpersonal distance in Figs. 4.3–4.4 (*Combined* set). I used Matlab's `ksdensity` function to generate these plots. In the case

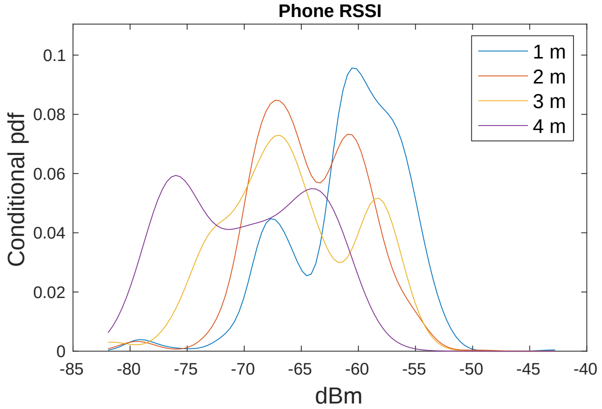
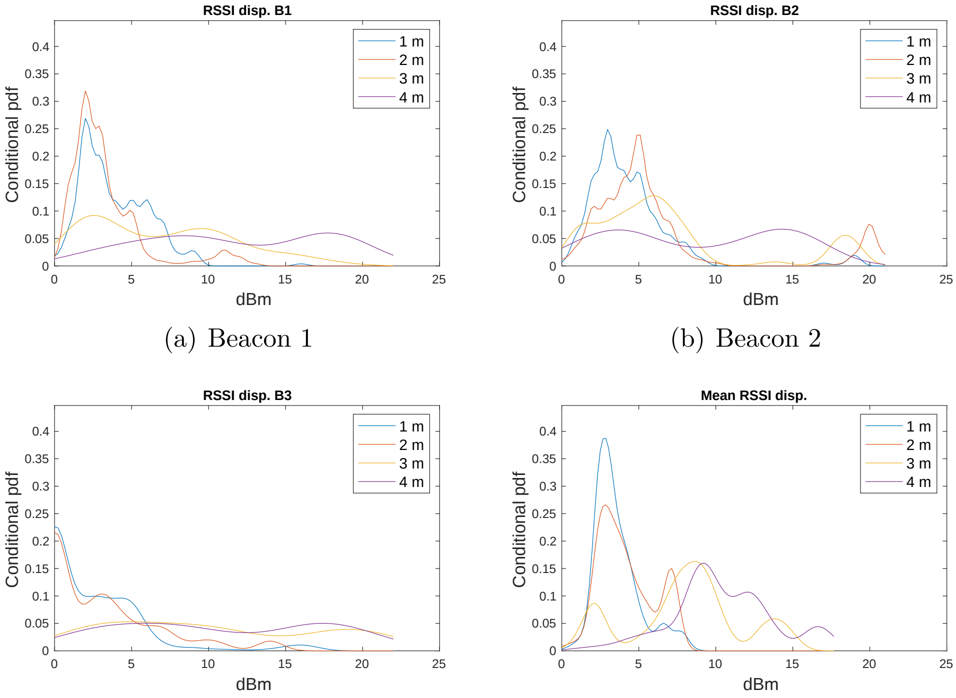


Figure 4.3: The pdf of the Phone RSSI, conditioned on the four different interpersonal distances considered (Home data collection).

of the Phone RSSI indicator, one would expect the mode of the conditional pdf to move leftwards (smaller RSSI values) for larger distances. From the plots in Fig. 4.3, it can be seen that the distributions of the Phone RSSI indicator at each distance are actually multimodal. As expected, the distributions allocate more mass towards lower RSSI values as the interpersonal distance increases, although a large overlap can be noticed for the pdf conditioned on distances of 2 and 3 meters.

In the case of individual RSSI disparity indices (Fig. 4.4), one can notice that the conditional pdf are relatively narrow and centered around small values for distances of 1 and 2 meters, while they become broader (larger variance) for distances of 3 and 4 meters. Indeed, only the disparity from Beacon 2 appears to be useful for discrimination between distances of 1 and 2 meters. The mean RSSI disparity index reflects this overall behavior, with the pdf conditioned on distances at 1 and 2 meters fairly well separated from those at 3 and 4 meters.

Given that both the phone RSSI index and the RSSI disparity index have distributions that correlate, to some extent, with interpersonal distances, it can be of interest to analyze the joint statistics of these two features. Specifically, I



(a) Beacon 1

(b) Beacon 2

(c) Beacon 3

(d) Mean RSSI

Figure 4.4: The pdf of the measured RSSI disparity, conditioned on the four different interpersonal distances considered (Home data collection). The last plot refers to the mean of the RSSI disparity over the three beacons.

consider the vector formed by phone RSSI and mean RSSI disparity, and study how the distribution of this vector conditioned on distances larger or smaller than a given threshold D_0. Fig. 4.5 displays the logarithm of the ratio of the joint pdf of this vector, conditioned on $D \leq D_0$ and on $D > D_0$, respectively with D_0 equal to 1 or 2 meters. Region with large positive or negative values of this quantity indicate good discriminability. These figures suggest that discrimination using individual indicators may be challenging, and that a 2-D classifier with non-separable boundaries may be called for. In this study, I considered a very simple linear classifier that assigns equal weights to the two features. In other words, this classifier applies a threshold to an index equal to the mean RSSI disparity minus the phone RSSI index (indicated as *Phone + RSSI disp.* in the figures). Large values of this index are likely to indicate large interpersonal distance.

ROC curves (plotting true positive rate, TPR, against false positive rate, FPR) are shown in Fig. 4.6. Each curve is obtained by varying a threshold on the considered measurements, where a value larger than the thresholds (or smaller, in the case of phone RSSI) indicates a distance $D > D_0$, for D_0 equal to 1 or 2 meters. The values of the area under the curve (AUC) for the relevant features are shown in Tab. 4.3, 4.4.

From Fig. 4.6, it is seen that the mean RSSI disparity performs better than individual RSSI disparities, at least for small values of FPR. The mean RSSI disparity index proved largely superior to the phone RSSI index in terms of distance discrimination for this data set. Depending on the distance threshold, the best results (in terms of AUC) are obtained by either by the mean RSSI disparity or by the phone+RSSI disparity feature. The effect of the distance threshold D_0 was relatively minor, with the best value at D_0= 1 meter obtained with the phone + RSSI disparity feature (AUC=0.86), and the best value at D_0= 2 meter obtained

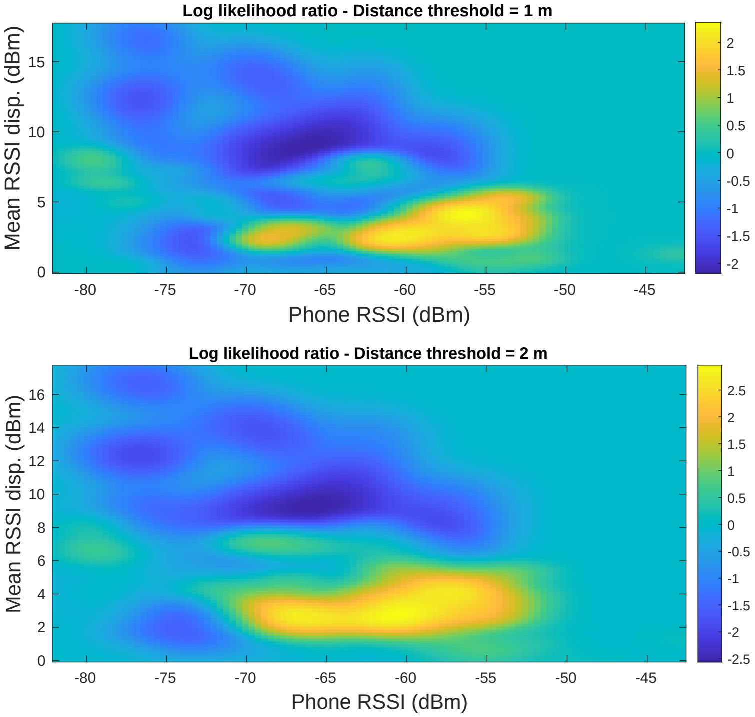


Figure 4.5: The log likelihood ratio of the vector formed by phone RSSI and mean RSSI disparity, conditioned on the interpersonal distances being smaller (null hypothesis) or larger (alternative hypothesis) than the considered threshold of 1 m or 2 m (Home data collection).

with the mean RSSI disparity feature (AUC=0.886).

It is important to note that I observed a large variance across data sets, even though all three data sets were acquired using similar modalities. For example, as shown in Tab. 4.4, data from Set 1 gave substantially worse discrimination results than for the other sets, especially when the distance threshold was set to D_0= 2 meters.

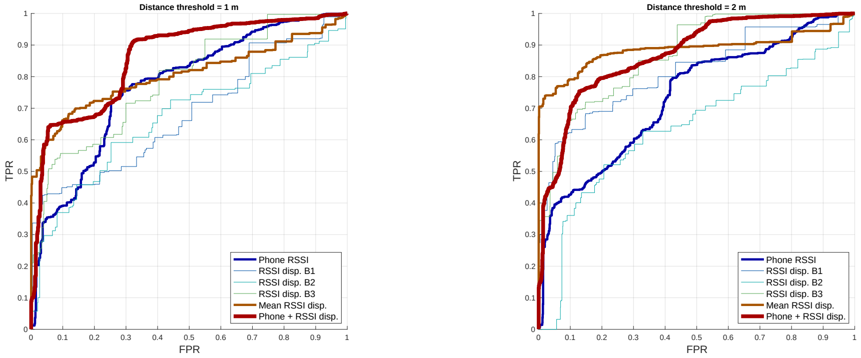


Figure 4.6: ROC curves using the considered measurements for proximity detection at distance thresholds of 1 m and 2 m (Home data collection).

A Realistic Case: Campus Shuttle Bus

Public transit is arguably one of the most appealing application scenarios for proximity detection. Social distancing may be difficult to observe inside a bus vehicle or a train car, which calls for mitigation measures based, among other things, on contact tracing. In addition, proximity measurements may be used to generate a crowdedness index, which could be broadcast to passengers waiting at bus stops or train stations. Passengers can then decide whether to board that bus vehicle or wait for the next one, or board a less crowded train car.

In order to evaluate the different considered features for proximity detection, I instrumented a bus shuttle vehicle in the campus with four BLE beacons, of the same type and with the same advertisement rate as in the previous case (see Fig. 4.7). Signal propagation is known to be complex in buses due to the existence of a strong radio signal reflector such as metal [95]. I conducted two data collection exercises, the first with the Power level of the beacons set to 1 (October 2020), the second, with power level set to 2 (February 2020). Power level is the strength of the signal that a beacon broadcasts. Signal strength (RSSI) is in decibels relative to a milliwatt (dBm). The maximum power that is available in iBeacons is 7 (4 dBm) that can be ranged up to 70 m. I set the power level to 1 (-20 dBm) and 2 (-16 dBm) that can be ranged approximately up to 4 m and 10 m. Note that while a higher emission power enables longer transmission distances, it also reduces the life time of a battery-operated beacon. With default power settings (power level equal to 3 or -20 dBM and advertisement interval set to $350ms$) the battery can last up to 2 years. Two experimenters used an iPhone 7 and an iPhone 8 for data collection from the sequence of seat pairs described in Tab. 4.1 and 4.2. Note that these sets contain a larger variety of interpersonal distances than for the Home data set, and that data for the same distance could come from multiple location pairs.

For each seat pair, both experimenters first collected data for two minutes while holding their phone in their hand, then for two minutes while keeping their phone in their front pants pocket. Data was collected while the vehicle was driven along its route, with passengers occasionally boarding and leaving the bus. At most three passengers were in the bus at the same time during data collection (note that, due to social distancing restrictions, at most six passengers were allowed in the vehicle at the same time.)

The log ratio of the joint pdf of the vector formed by phone RSSI and mean RSSI disparity, conditioned on distances larger and smaller, respectively, of a

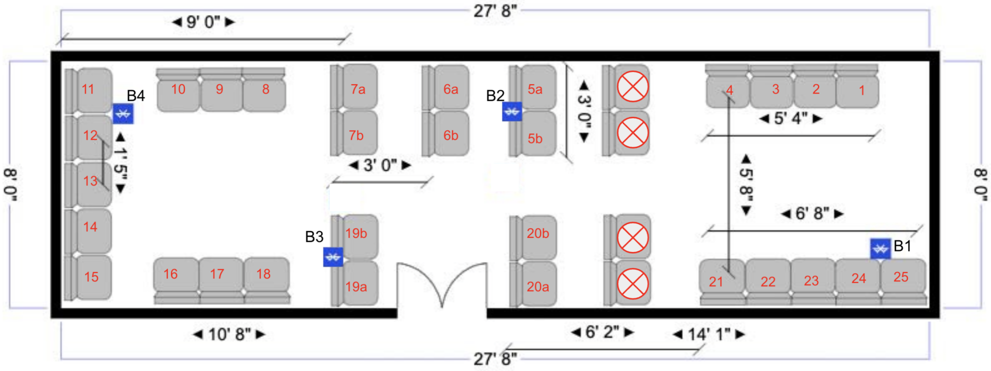


Figure 4.7: Shuttle data collection layout representing the location of the beacons deployed on the shuttle ceiling and seat numbers. Right of the figure is the front of the shuttle. Crossed red circle seats are blocked seats for COVID-19 distancing.

Distance (m)	Seat pairs
0.5	(4, 3), (7a, 8), (19b, 19a), (18, 19a)
1.0	(4, 2)
1.8	(7a, 19a), (4, 21)
2.5	(4, 20a), (4, 25), (5a, 19a)
3.28	(7a, 15)
4.0	(4, 19a)

Table 4.1: Power level 1 data collection seat pairs

Distance (m)	Seat pairs
0.5	(4,3), (7a, 7b), (20b, 20a), (18, 19a), (5a, 6a)
1.0	(4, 2)
1.5	(7a, 19b), (11, 9)
1.8	(7a, 19a), (11, 15), (4, 21), (4, 5a), (21, 25), (5a, 7a), (5a, 20a)
2.5	(4, 20a), (4, 25), (7a, 20a), (7a, 11)
3.28	(7a, 15)
4.0	(4, 19a)
4.88	(11, 20a)
6.0	(4, 11)

Table 4.2: Power level 2 data collection seat pairs

threshold D_0, are shown shown in Fig. 4.8 for D_0=1 meter and 2 meters. As in the Home data collection case, I compiled data at identical interpersonal distance into the same set (*Combined*). I should note that, unlike the House data collection, various occluding surfaces (such as the backs of the vehicle seats) affected signal transmission even for small interpersonal distances (except when the participants were sitting next to each other.) This may be one of the reasons for the broad distribution of the phone RSSI values when $D \leq D_0$ (ranging from -80 dBm to -35 dBm).

ROC curves for the considered features are shown in Fig. 4.9 for the Combined set and D_0=1 meter and 2 meters. AUC values for the individuals sets (at different BLE power level) as well for the combined case are shown in Tab. 4.5 and 4.6. Performances were substantially inferior to the those obtained with the Home data set, especially for $D_0 = 1$ meter. In all cases, the best results were seen using either mean RSSI disparity, or the phone + RSSI disparity feature. However, the improvement with respect to the phone RSSI was relatively marginal.

As in the case of the Home data collection, I observed a large variation of performance using phone RSSI for the two individual sets at different BLE power levels. This is somewhat baffling, considering that the BLE power from the beacons level should not affect the received RSSI from another phone. It may be that other uncontrolled factors (e.g., the presence of other passengers) may have contributed to this discrepancy. Interestingly, better results were obtained for the mean RSSI disparity feature using power level 1 than for power level 2 (see also the ROCs shown in Fig. 4.10.)

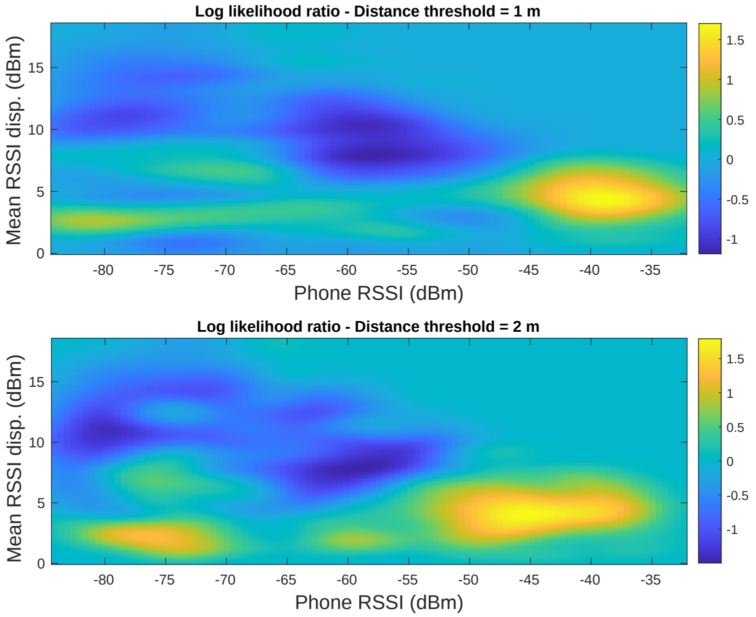


Figure 4.8: The log likelihood ratio of the vector formed by phone RSSI and mean RSSI disparity, conditioned on the interpersonal distances being smaller (null hypothesis) or larger (alternative hypothesis) than the considered threshold of 1 m or 2 m (Shuttle data collection at BLE power level 1 and 2).

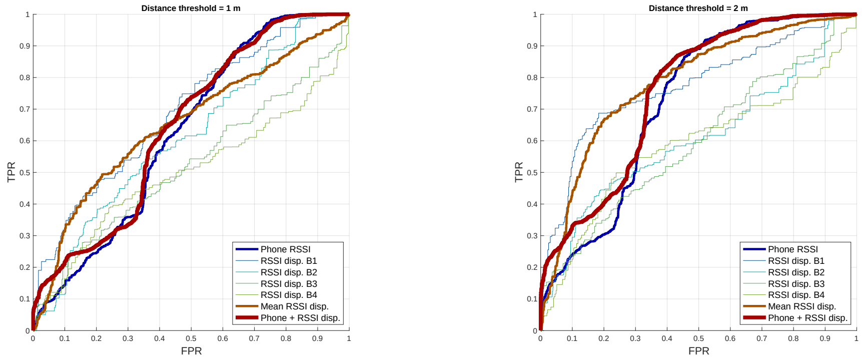


Figure 4.9: ROC curves using the considered measurements for proximity detection at distance thresholds of 1 m and 2 m (Shuttle data collection at BLE power level 1 and 2).

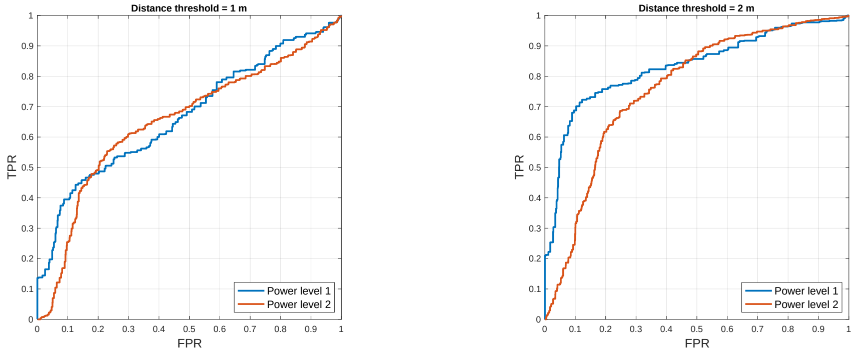


Figure 4.10: ROC curves using mean RSSI disparity for proximity detection at distance thresholds of 1 m and 2 m. Shuttle data collection at BLE power level 1 (blue curve) and at BLE power level 2 (red curve).

AUC (D=1m)	**Set 1**	**Set 2**	**Set 3**	**Combined**
Phone RSSI	0.680	0.793	0.783	0.770
Mean RSSI Disp.	0.605	0.883	0.733	0.805
Phone + RSSI Disp.	**0.708**	**0.915**	**0.876**	**0.860**

Table 4.3: AUC results for Home data sets when the distance threshold was set to D_0= 1 meter. Best results for each column are shown in boldface.

AUC (D=2m)	**Set 1**	**Set 2**	**Set 3**	**Combined**
Phone RSSI	0.588	0.72	0.834	0.735
Mean RSSI Disp.	**0.689**	**0.987**	0.755	**0.886**
Phone + RSSI Disp.	0.648	0.909	**0.923**	0.872

Table 4.4: AUC results for Home data sets when the distance threshold was set to D_0= 2 meters.

AUC (D=1m)	**Pwr. level 1**	**Pwr. level 2**	**Combined**
Phone RSSI	0.492	0.720	0.626
Mean RSSI Disp.	**0.670**	0.653	**0.652**
Phone + RSSI Disp.	0.568	**0.727**	0.646

Table 4.5: AUC results for Shuttle data sets when the distance threshold was set to D_0= 1 meter.

AUC (D=2m)	**Pwr. level 1**	**Pwr. level 2**	**Combined**
Phone RSSI	0.595	0.772	0.720
Mean RSSI Disp.	**0.831**	0.760	**0.783**
Phone + RSSI Disp.	0.698	**0.782**	0.754

Table 4.6: AUC results for Shuttle data sets when the distance threshold was set to D_0= 2 meters.

4.2.4 Discussion and Conclusions

I have presented a comparative analysis of RSSI–based techniques for proximity detection in different environments. My goal was to assess how well different types of measurements (namely, direct RSSI measurement from another user's phone, and disparity of RSSI signals received from multiple BLE beacons) can discriminate between interpersonal distances of 1 meter and 2 meters. The results show that the mean RSSI disparity index performs as well or better than the direct phone RSSI index for these tasks, and that a simple combination of the two indices often produces the best results.

While the experimental setups were effective for data collection (and, in the case of the Shuttle data set, representative of real-world conditions), this study has a number of limitations. I only used two models of one smartphone brand (iPhone). It is well known [96] that different smartphones have different characteristics in terms of the received signals from a BLE beacon, and a more exhaustive study with multiple smartphone brands would be necessary before these results can be generalized. My “toy case" of an instrumented living room may not be representative of a more complex and crowded environment such as a shopping mall or an office building. It is likely that different BLE beacon placements would result in widely different performances, and I plan to experiment with different placement layouts in the future. Only few passengers were present during the study, and I am planning for a more extensive data gathering once social distancing rules are relaxed and more students will be using my campus transportation system. The use of more sophisticated mechanisms than the linear classifier used in the tests (see e.g. [84, 89]) could be considered, provided that enough data is collected to enable good generalization.

4.3 Effortless Ticketing and Crowdedness Monitoring

Effortless (or implicit [6]) ticketing, refers to methods that enable payment of the correct fare as triggered by the mere presence of the user inside the vehicle. Current touchless fare payment technology still requires travelers to approach a near-field communication (NFC) reader or possibly a QR reader [7] located in the vehicle. This creates “accumulation points” of social proximity, which may slow the flow of passengers entering the vehicle and thus increase boarding times. Increasingly, agencies are offering the possibility to purchase tickets through an app (e.g., Transit, or Google Maps). Although this obviates crowding near card readers in the vehicle, it requires travelers to identify the correct fare (e.g., if the cost depends on the fare zones traversed), or to input the intended route in the app. A real effortless ticketing system would not require the users to take any actions, except for starting an app in their smartphone. It would automatically identify the vehicle boarded by the passenger (Be In/Be Out, or BIBO, modality [6]), and charge the correct fare. Users would not need to input information such as their itinerary or the specific bus or train line they are going to use. Upon boarding the bus vehicle or train car, the user would receive a notification (e.g., via a vibration) from the system that the vehicle has been identified, and that that ticketing is taken care of.

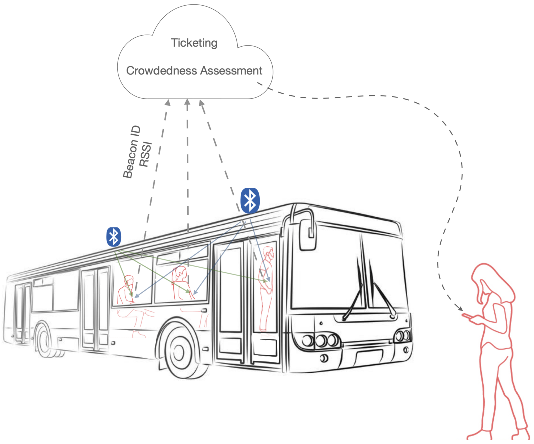


Figure 4.11: Our envisioned system uses RSSI data recorded from the in-vehicle BLE beacons for passenger positioning, enabling effortless fare payment and crowdedness assessment.

The same mechanism that enables effortless ticketing can be used to assessing and track the distribution of passengers in a vehicle. Crowdedness monitoring has received increased recent attention [8]. I envision a system that measures not only the approximate number of passengers, but also their spatial distribution in the vehicle. This could be very useful when deciding which door to enter a vehicle from. For example, if riders in a train cart or bus vehicle are concentrated in the front half, a passenger waiting at the stop may decide to enter from the back door (see Fig. 4.11). This information could also be very valuable for transit agencies, which can put in place provisions to ensure a uniform distribution of passengers in their vehicles. A few transit apps provide this occupancy information (when available) to passengers awaiting at a bus stop. Passengers can then choose, based on this information, whether to board the upcoming vehicle, or, if they determine that the vehicle is too crowded for their comfort level, wait for the next one, or use a different means of transportation. Crowdedness can be measured using specialized sensors (e.g., seat sensors or cameras), or through crowdsourcing [97,98]. Occupancy sensors, however, are generally expensive, involve some form of vehicle retrofitting, entail maintenance costs, and typically require an additional data communication channel. Crowdsourcing approaches are attractive because they require no instrumentation, but they depend on the willingness of passengers to input data during a trip. It has been observed that contributors to crowdsourcing projects (e.g., OpenStreetMap) tend to belong to the more affluent and educated portion of the society [99], which may not be representative of large swaths of the population riding public transit.

I propose to use Bluetooth Low Energy (BLE) beacons as the underlying technology for both services considered (effortless ticketing and crowdedness monitoring). BLE beacons are inexpensive and unobtrusive. Battery operated models (e.g., Kontakt Tough Beacon TB18-2) can last up to 80 months on a battery charge, and require no vehicle retrofitting (including wiring) nor maintenance during this period. These factors are critically important, given the tremendous budget constraints that agencies are experiencing due to recent loss of ridership.

Our concept is very simple. Passengers start an app on their phone; once they board the bus, light rail, or train vehicle, the app detects the ID of the onboard BLE beacons, as well as the Received Signal Strength Indicator (RSSI) from each beacon. This information is transmitted (by the user's phone) to a cloud server, which is cognizant of the association between beacon ID and specific trip in the agency's General Transit Feed Specification (GTFS) table [100]. The system can then charge the user the appropriate fare, and also (based on the received RSSI), determine the location of the user in the vehicle. By aggregating information from multiple users in the same vehicle, a measure of crowdedness and of its spatial distribution is generated, which can be advertised through standard mechanism, such as GTFS Real Time [101], for other online transit services or apps to be picked up.

In this contribution, I present results from a preliminary study on passenger localization within a bus vehicle using BLE beacons. I instrumented a campus shuttle bus with four BLE beacons, and conducted multiple data collection session. RSSI data was collected from all beacons while the experimenter sat in different location within the vehicle. Analysis of this data confirms that localization accuracy of up to 1 meter (along the length of the bus) can be achieved using BLE beacons in realistic conditions.

4.3.1 Related Work

The potential of BLE beacons for effortless/implicit ticketing using the BIBO paradigm was first demonstrated by Narzt and colleagues in 2015 [6]. In their prototype, the passengers' smartphones were tasked with sending (via BLE) an ID that was received and processed by a system within the vehicle. This approach, however, requires some level of retrofitting (including wiring and Internet connectivity) that may discourage adoption by cash-strapped agencies. The system recently proposed by Ferreira et al. [102] (developed through a participatory co-design cycle involving potential customers [103]) is closer to this envisioned BLE beacon placement scenario. This system, however, was only tested with a single beacon placed in one bus vehicle.

While BLE beacons, as well as NFC or QR code readers, can be used to monitor the presence of passengers in a vehicle, other mechanisms have been explored that don't require such sensors. For example, by matching the GPS tracks [104] or the time series from inertial or barometric sensors collected by the user's smartphones [105–107] with those recorded by a sensor in the bus vehicle, it is possible to determine whether the user is on a certain bus route. However, GPS-denied environments, or spurious motion of the smartphone, can generate errors. None of these methods can provide information about the location of the user in the vehicle, which is necessary to compute the spatial distribution of passengers.

The use of BLE beacons for localization has been well studied. In ideal conditions, power decay models [108] could be used in a multilateration scheme to precisely compute the location of the receiver (the user's smartphone). In practice, researchers have found that power decay models cannot be relied on, due to a multiplicity of reasons including multi-path fading and variations in time of the signal power [109]. The standard approach is based on fingerprinting [110], whereby RSSI data is collected from a dense set of known locations, and the user's location is then regressed from the received RSSI vector.

4.3.2 In–Vehicle Positioning

The goal of this study was to assess the performance of a positioning system based on the RSSI from multiple BLE beacons placed in a bus vehicle. Note that various factors could complicate the location inference problem, such as multiple reflections, occlusions by obstacles (e.g. the seat backs) or other passengers, as well as self-occlusions (e.g., the user keeping the smartphone tucked in a pocket.) In order to ascertain whether localization in a bus vehicle is even possible with data from BLE beacons, I conducted several data collection sessions in a campus shuttle bus. The vehicle (8.3 meters long, 2.6 meters wide) was equipped with four Kontakt Tough TB15-1 BLE beacons, configured as iBeacons with an advertisement interval of 350 ms (see Fig. 4.12.) I created an iPhone app that collects timestamped RSSI data from the different beacons. During each data collection session, an experimenter sat on different seats as the vehicle drove through its regular route, while collecting data from the BLE beacons using either an iPhone 7 or an iPhone 8. At each seat, the experimenter first recorded data for two minutes while holding the phone in their hand, then for two minutes while keeping the phone in their front pant pocket. The vehicle was empty for most of the time, except for a few occasional passengers (at most three passengers besides the experimenter at a time).

I first ran three data collection sessions with the BLE beacons set to Power level 1. With this setting, the RSSI at 1 meter of distance is of -84 dBm, and the nominal range is of approximately 4 meters. I then conducted one session at Power level set to 2 (RSSI of -81dBm at 1 meter, nominal range of approximately 10 meters). Figs. 4.13 and 4.14 show the layout of the beacons in the bus, as well as the seats considered for data collection for each power level. This data set was used to ascertain whether it would be possible to estimate, based on the RSSI received from different beacons, the location of a user across the length of the bus (i.e., to determine, at least approximately, the seat row in which the user was positioned.) I did not attempt to estimate the user's position across the width of the vehicle, given the vehicle's relatively narrow geometry.

The RSSI collected at each seat, averaged over time, over the two phone placements (in hand and in pocket), and, for the case of Power level 1, over the three sessions, is presented in the top plot of the figures. In this plot, the horizontal axis represents the seat row position along the length of the bus. For each seat, I show the average RSSI received from each BLE beacon. As expected, the RSSI from beacon B1 (in front of the vehicle) was generally higher for seats in the front half of the vehicle. The opposite behavior can be observed for the data received from beacon B4, at the back of the vehicle. The RSSI data collected from B2 and B3 had a less clear dependence on the seat location. Interestingly, signal was received even at seats that were at about 7 meters of distance from a beacon even when the BLE power level was set to 1 (with a nominal 4 meter range). This is likely due to reflection from the metallic surfaces of the vehicle.

To verify whether the vector of RSSI data collected from the different beacons could be used to accurately measure the seat row location of the user, I trained a simple linear predictor of seat row position from data collected in half of the seats (shown with a dark contour in the figures.) I then used this predictor to estimate the row location of the remaining seats, based on the recorded average RSSI data. Here is the model assuming using two beacons data. $B_i = [B_1(i), B_2(i)]$ is the RSSI from the two beacons at seat i. $I(i)$ is the true 1-D position of seat i. The model is $I_{est(i)} = Alpha + Beta * B(i)^T$, where $Beta = [Beta_1, Beta_2]$. I find Alpha and Beta by least squares regression from half of the seats, and apply it to the $B(i)$ of the other half to compute their I_{est}. Then I compare I_{est} for test seats with the ground-truth I to get the error.

The results are shown in the lower row of Figs. 4.13 and 4.14, with seat identified by their color. Using data collected from all four beacons, the root mean square error (RMSE) of estimated row position was of 0.57 meters (max error: 1.1 m) when using Power level 1, and of 0.33 meters using Power level 2 (max error: 0.62 meters).

As noted earlier, the plot of the RSSI values in Figs. 4.13 and 4.14 suggests that while data from beacons B1 and B4 clearly correlates with the seat row location, the remaining beacons appear to be less informative. Based on this observation, I repeated the same test, but only considering RSSI data collected from B1 and B4. In this case, the RMSE of estimated row position was of 0.50 meters (max error: 0.93 m) when using Power level 1, and 0.30 meters using Power level 2 (max error: 0.66 meters).

From this preliminary analysis, I can draw the following observations:

1. Localization of a passenger within the length of the bus vehicle is possible using BLE beacons, to at least 1 meter accuracy. This information could be used to derive a coarse–scale crowdedness index. For example, one could divide the length of the bus into 3 or 4 section, and count the number of passengers in each section (where, as in the system envisioned in Sec. 4.1, the RSSI data would be transmitted from the users' phones to a cloud server, e.g. as part of an effortless payment app.)

2. Setting the beacons at power level 2 appears to produce better localization accuracy. It should be noted, though, that a higher transmission power directly affects the lifetime of the beacon when battery operated. The trade-off between accuracy and system lifetime needs to be carefully considered when designing a real-world BLE beacons system.

3. Using two beacons (one in front of the bus, and one in the back) appears to give similar (or better) results than using data from four beacons, distributed along the length of the bus. This somewhat surprising results suggests that data from beacons B2 and B3, which were placed in the middle of the bus, contributed little (and possibly noisy) information for the purpose of positioning.

4.3.3 Discussion and Conclusions

The data collection and analysis described in the previous section has shown promising results for the use of BLE beacons in measuring the position of a passenger along the length of the vehicle. However, more research work is needed to obtain a system that is robust and reliable in the face of multiple adversarial situations. For example, while the radio signal from one or multiple beacons can almost certainly be detected once one has boarded the vehicle, the same signal could potentially be detected also outside the vehicle. This could create false alarms, for example when a passenger is waiting at a stop, and a bus or train the passenger is not planning to board is coasting to the stop. These situations could be managed by looking at the time series of RSSI measurements, possibly combined with information from the inertial sensors in the user's smartphone. For example, if the sensors detect that the user is moving of motion that is consistent with that of a vehicle [111], and connection with the beacons remains stable, it could be safely assumed that the user has boarded the departed bus. Although a similar result could be obtained by matching the GPS track of the user's smartphone with that of the bus, although this would not be an option in a GPS-denied environment (e.g., in a subway).

Another situation that could potentially generate errors is one with multiple bus vehicles arriving at the stop at the same time. In this case, it could be possible that the user's smartphone, even after boarding, may receive radio signal from beacons in other nearby vehicles, potentially triggering an erroneous system response. Even in this case, joint analysis of inertial and RSSI measurements could break the ambiguity and assign the passenger to the correct vehicle.

Standard fingerprinting procedures are unlikely to produce reliable results unless confounding factors such as the presence of nearby travelers, whether the user is standing or sitting, and whether the user is holding the phone in their hand or tucked in a pocket, are taken into account. I believe that addressing the open problems mentioned above is only possible if a representative data set, collected in realistic situations, and adequately annotated, is made available. A number of open access data sets containing data from BLE beacons (for indoor localization applications; e.g., [112, 113]) or from inertial sensors (e.g., [114–117]) already exist. However, none of these data sets would be representative of the situations considered here. What is needed is a collection of synchronized measurements of RSSI, inertial data, and GPS tracks, collected from passengers' smartphones, that could be analyzed viz-a-viz the known trajectory of the transit vehicle that was boarded by these users. This data must be recorded by multiple different users, using different types of smartphones, and in various different conditions (location in the vehicle, crowdedness level, atmospheric conditions).

I have presented results from a preliminary study of a system that can localize passengers in a bus vehicle from the RSSI signal received from multiple BLE beacons placed in the vehicle. In spite of the non-ideal conditions of this environment (with multiple reflections and occlusions), a simple linear predictor was shown to produce better than 1 meter accuracy. This simple experiment suggests that coarse-scale localization within the length of the bus is possible using BLE beacons. This localization system could be used in the context of effortless ticketing and crowdedness assessment applications.

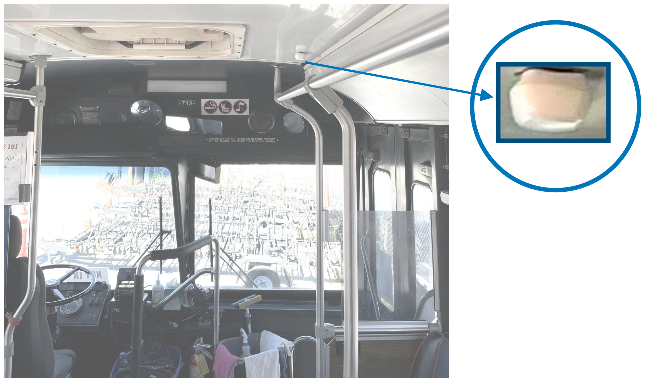


Figure 4.12: Example of placement of a BLE beacon (Kontakt Tough TB15-1) on the ceiling of the shuttle bus.

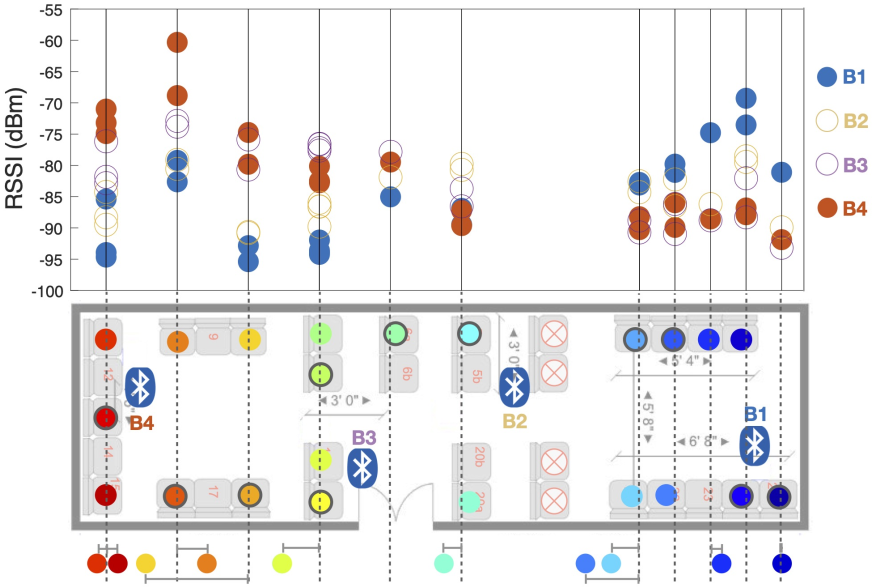


Figure 4.13: Layout of BLE beacons in shuttle bus (the front of the vehicle is to the right of the figure.) The seats from which RSSI data was collected are marked with distinctive colors. The linear prediction model was trained from data collected at the seats marked by a dark contour. The average RSSI values received at each seat are displayed in the top plot for each BLE beacon. The horizontal axis represents the seat row location along the length of the bus. The bottom row shows the seat row location estimated from RSSI data using the linear predictor (only seats that were not used in training are shown.) Errors (estimated vs. actual seat row location) are shown by gray segments. The power level of the BLE beacons was set to 1.

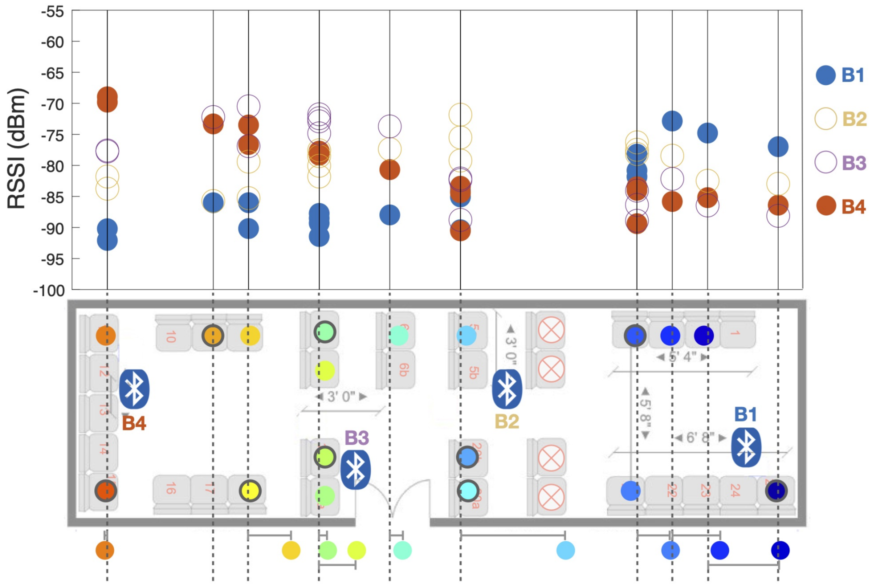


Figure 4.14: See caption of Fig. 4.13. The power level of the BLE beacons was set to 2.

Bibliography

[1] Y. Wang, G. Huynh, and C. Williamson, "Integration of google maps/earth with microscale meteorology models and data visualization," *Computers & Geosciences*, vol. 61, pp. 23–31, 2013.

[2] A. Compagno, J. Facchinello, J. Mejias, and P. Perez, "Indoor navigation system with beacons," 2014.

[3] F. Mirzaei, R. Manduchi, and S. Kurniawan, "Public transit accessibility: Blind passengers speak out," in *International Conference on Computers Helping People with Special Needs.* Springer, 2018, pp. 277–282.

[4] N. R. Jones, Z. U. Qureshi, R. J. Temple, J. P. Larwood, T. Greenhalgh, and L. Bourouiba, "Two metres or one: what is the evidence for physical distancing in COVID-19?" *bmj*, vol. 370, 2020.

[5] P. Bahl, C. Doolan, C. De Silva, A. A. Chughtai, L. Bourouiba, and C. R. MacIntyre, "Airborne or droplet precautions for health workers treating COVID-19?" *The Journal of infectious diseases*, 2020.

[6] W. Narzt, S. Mayerhofer, O. Weichselbaum, S. Haselböck, and N. Höfler, "Be-In/ Be-Out with Bluetooth Low Energy: Implicit ticketing for public transportation systems," in *IEEE 18th International Conference on Intelligent Transportation Systems*, 2015.

[7] "Mobile ticketing: Why barcode?" 2019. [Online]. Available: https://www.masabi.com/2019/10/28/mobile-ticketing-why-barcode/.

[8] M. Wanek-Libman, "Occupancy data: A real crowd pleaser," 2020-09-24. [Online]. Available: https://www.masstransitmag.com/technology/passenger-info/article/21154973/occupancy-data-a-real-crowd-pleaser

[9] J. Paek, J. Ko, and H. Shin, "A measurement study of ble ibeacon and geometric adjustment scheme for indoor location-based mobile applications," *Mobile Information Systems*, vol. 2016, 2016.

[10] D. Yoo, J. Zimmerman, A. Steinfeld, and A. Tomasic, "Understanding the space for co-design in riders' interactions with a transit service," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, 2010, pp. 1797–1806.

[11] R. G. Golledge, J. R. Marston, and C. M. Costanzo, "Attitudes of visually impaired persons toward the use of public transportation," 2001.

[12] S. Azenkot, S. Prasain, A. Borning, E. Fortuna, R. E. Ladner, and J. O. Wobbrock, "Enhancing independence and safety for blind and deaf-blind public transit riders," in *Proceedings of the SIGCHI conference on Human Factors in computing systems.* ACM, 2011, pp. 3247–3256.

[13] M. Banâtre, P. Couderc, J. Pauty, and M. Becus, "Ubibus: Ubiquitous computing to help blind people in public transport," in *International Conference on Mobile Human-Computer Interaction.* Springer, 2004, pp. 310–314.

[14] M. Z. H. Noor, I. Ismail, and M. F. Saaid, "Bus detection device for the blind using rfid application," in *2009 5th International Colloquium on Signal Processing & Its Applications.* IEEE, 2009, pp. 247–249.

[15] R. Jacob, B. Shalaik, A. C. Winstanley, and P. Mooney, "Haptic feedback for passengers using public transport," in *International Conference on Digital Information and Communication Technology and Its Applications.* Springer, 2011, pp. 24–32.

[16] J. Kostiainen, C. Erkut, and F. B. Piella, "Design of an audio-based mobile journey planner application," in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments.* ACM, 2011, pp. 107–113.

[17] G. Flores and R. Manduchi, "A public transit assistant for blind passengers: Development and experiments," 2018.

[18] K. Hara, S. Azenkot, M. Campbell, C. L. Bennett, V. Le, S. Pannella, R. Moore, K. Minckler, R. H. Ng, and J. E. Froehlich, "Improving public transit accessibility for blind riders by crowdsourcing bus stop landmark locations with google street view: An extended analysis," *ACM Transactions on Accessible Computing (TACCESS)*, vol. 6, no. 2, p. 5, 2015.

[19] C. P. Tribby and P. A. Zandbergen, "High-resolution spatio-temporal modeling of public transit accessibility," *Applied Geography*, vol. 34, pp. 345–355, 2012.

[20] M. Campbell, C. Bennett, C. Bonnar, and A. Borning, “Where’s my bus stop?: supporting independence of blind transit riders with stopinfo,” in *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility.* ACM, 2014, pp. 11–18.

[21] T.-h. Kim, C. Ramos, and S. Mohammed, “Smart city and iot,” 2017.

[22] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global positioning system: theory and practice.* Springer Science & Business Media, 2012.

[23] S. Madry *et al.*, *Global navigation satellite systems and their applications.* Springer, 2015.

[24] Y. Gwon and R. Jain, “Error characteristics and calibration-free techniques for wireless lan-based location estimation,” in *Proceedings of the second international workshop on Mobility management & wireless access protocols.* ACM, 2004, pp. 2–9.

[25] K. Chintalapudi, A. Padmanabha Iyer, and V. N. Padmanabhan, “Indoor localization without the pain,” in *Proceedings of the sixteenth annual international conference on Mobile computing and networking.* ACM, 2010, pp. 173–184.

[26] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, and E. Steinbach, “Graph-based data fusion of pedometer and wifi measurements for mobile indoor positioning,” in *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing.* ACM, 2014, pp. 147–158.

[27] P. Bahl, V. N. Padmanabhan, V. Bahl, and V. Padmanabhan, “Radar: An in-building rf-based user location and tracking system,” 2000.

[28] B. F. D. Hähnel and D. Fox, “Gaussian processes for signal strength-based location estimation,” in *Proceeding of robotics: science and systems*, 2006.

[29] 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.

[30] S. He and S.-H. G. Chan, “Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490, 2015.

[31] J. Biswas and M. Veloso, “Wifi localization and navigation for autonomous indoor mobile robots,” in *2010 IEEE international conference on robotics and automation.* IEEE, 2010, pp. 4379–4384.

[32] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *IEEE journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, 2015.

[33] M. Murata, D. Ahmetovic, D. Sato, H. Takagi, K. M. Kitani, and C. Asakawa, "Smartphone-based localization for blind navigation in building-scale indoor environments," *Pervasive and Mobile Computing*, vol. 57, pp. 14–32, 2019.

[34] F. Zafari, I. Papapanagiotou, M. Devetsikiotis, and T. Hacker, "An ibeacon based proximity and indoor localization system," *arXiv preprint arXiv:1703.07876*, 2017.

[35] J. Budina, O. Klapka, T. Kozel, and M. Zmítko, "Method of ibeacon optimal distribution for indoor localization," in *International and Interdisciplinary Conference on Modeling and Using Context.* Springer, 2015, pp. 105–117.

[36] M. Castillo-Cara, J. Lovón-Melgarejo, G. Bravo-Rocca, L. Orozco-Barbosa, and I. García-Varea, "An analysis of multiple criteria and setups for bluetooth smartphone-based indoor localization mechanism," *Journal of Sensors*, vol. 2017, 2017.

[37] A. T. Parameswaran, M. I. Husain, S. Upadhyaya *et al.*, "Is rssi a reliable parameter in sensor localization algorithms: An experimental study," in *Field failure data analysis workshop (F2DA09)*, vol. 5. IEEE, 2009.

[38] T. Conceição, F. N. dos Santos, P. Costa, and A. P. Moreira, "Robot localization system in a hard outdoor environment," in *Iberian Robotics conference.* Springer, 2017, pp. 215–227.

[39] P. Kriz, F. Maly, and T. Kozel, "Improving indoor localization using bluetooth low energy beacons," *Mobile Information Systems*, vol. 2016, 2016.

[40] J. Rezazadeh, R. Subramanian, K. Sandrasegaran, X. Kong, M. Moradi, and F. Khodamoradi, "Novel ibeacon placement for indoor positioning in iot," *IEEE Sensors Journal*, vol. 18, no. 24, pp. 10 240–10 247, 2018.

[41] D. Sato, U. Oh, J. Guerreiro, D. Ahmetovic, K. Naito, H. Takagi, K. M. Kitani, and C. Asakawa, "Navcog3 in the wild: Large-scale blind indoor navigation assistant with semantic features," *ACM Transactions on Accessible Computing (TACCESS)*, vol. 12, no. 3, p. 14, 2019.

[42] M. Murata, D. Ahmetovic, D. Sato, H. Takagi, K. M. Kitani, and C. Asakawa, "Smartphone-based indoor localization for blind navigation across building complexes," in *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom).* IEEE, 2018, pp. 1–10.

[43] F. Zafari and I. Papapanagiotou, “Enhancing ibeacon based micro-location with particle filtering,” in *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2015, pp. 1–7.

[44] A. C. J. Malar, G. Kousalya, and M. Ma, “Markovian model based indoor location tracking for internet of things (iot) applications,” *Cluster Computing*, pp. 1–8, 2017.

[45] L. Ran, S. Helal, and S. Moore, “Drishti: an integrated indoor/outdoor blind navigation system and service,” in *Second IEEE Annual Conference on Pervasive Computing and Communications, 2004. Proceedings of the.* IEEE, 2004, pp. 23–30.

[46] J. Spilker Jr, “Gps signal structure and performance, global positioning system, navigation,” *Journal of the Institute of Navigation*, vol. 25, pp. 121–146, 1980.

[47] C. Rizos, “Making sense of the gps techniques, chapter 11 in manual of geospatial science and technology, j. bossler, j. jenson, r. mcmaster & c. rizos,” 2002.

[48] M. G. Wing, A. Eklund, and L. D. Kellogg, “Consumer-Grade Global Positioning System (GPS) Accuracy and Reliability,” *Journal of Forestry*, vol. 103, no. 4, pp. 169–173, 06 2005.

[49] M. Crossley, “A guide to coordinate systems in great britain,” *Ordnance Survey*, 1999.

[50] C. Papamanthou, F. P. Preparata, and R. Tamassia, “Algorithms for location estimation based on rssi sampling,” in *International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics.* Springer, 2008, pp. 72–86.

[51] M. S. Aman, H. Jiang, C. Quint, K. Yelamarthi, and A. Abdelgawad, “Reliability evaluation of ibeacon for micro-localization,” in *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON).* IEEE, 2016, pp. 1–5.

[52] H.-s. Cho, J. Ji, Z. Chen, H. Park, and W. Lee, “Accurate distance estimation between things: A self-correcting approach,” *Open Journal of Internet Of Things (OJIOT)*, vol. 1, no. 2, pp. 19–27, 2015.

[53] R. Pinto, F. N. Santos, and A. J. Sousa, “Robot self-localization based on sensor fusion of gps and ibeacons measurements,” in *11th edition of the Doctoral Symposium in Informatics Engineering (DSIE-16)*, 2016.

[54] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics.* MIT press, 2005.

[55] Kontakt.io. (2019). [Online]. Available: https://kontakt.io

[56] R. L. Klatzky, J. R. Marston, N. A. Giudice, R. G. Golledge, and J. M. Loomis, "Cognitive load of navigating without vision when guided by virtual sound versus spatial language." *Journal of experimental psychology: Applied*, vol. 12, no. 4, p. 223, 2006.

[57] M. Hossain, K. M. Qaiduzzaman, M. Rahman *et al.*, "Sightless helper: an interactive mobile application for blind assistance and safe navigation," in *International Conference on Cyber Security and Computer Science.* Springer, 2020, pp. 581–592.

[58] J. Brooke, "Sus: a "quick and dirty'usability," *Usability evaluation in industry*, vol. 189, no. 3, 1996.

[59] ... (2022) Blindsquare app. [Online]. Available: https://blindsquare.com

[60] ——. (2022) Blindways app. [Online]. Available: https://www.perkins.org/perkins-launches-blindways-app/

[61] A. Alvarado, A. Chong, Y. Kojitani, E. Orellana, E. Vadai, L. Zhang, G. Flores, R. Manduchi, and E. Miller, "RouteMe2: A cloud-based infrastructure for assisted transit," in *Transportation Research Board Annual Meeting*, Washington, DC, 2018.

[62] S. Madsen and L. Nielsen, "Exploring persona-scenarios-using storytelling to create design ideas," in *IFIP working conference on human work interaction design.* Springer, 2009, pp. 57–66.

[63] C. Van der Lelie, "The value of storyboards in the product design process," *Personal and ubiquitous computing*, vol. 10, no. 2, pp. 159–162, 2006.

[64] T. Snyder, "The pandemic could devastate mass transit in the U.S. – and not for the reason you think," *Politico*, 2021-01-21.

[65] L. Liu, H. J. Miller, and J. Scheff, "The impacts of COVID-19 pandemic on public transit demand in the United States," *Plos one*, vol. 15, no. 11, p. e0242476, 2020.

[66] M. Joselow, "There is little evidence that mass transit poses a risk of coronavirus outbreaks," *Scientific American*, vol. July, 2020.

[67] H. Clark, *Who Rides Public Transportation.* American Public Transportation Association, 2017.

[68] J. Kellett, R. Barreto, A. V. D. Hengel, and N. Vogiatzis, “How might autonomous vehicles impact the city? the case of commuting to central Adelaide,” *Urban policy and research*, vol. 37, no. 4, pp. 442–457, 2019.

[69] J. Tumlin, “Washington priorities for driverless cars,” 2021-02-17. [Online]. Available: https://thehill.com/blogs/congress-blog/technology/539140-washington-priorities-for-driverless-cars

[70] *National Academies of Sciences, Engineering, and Medicine 2021, "COVID-19 Addendum to Critical Issues in Transportation.* The National Academies Press, 2021.

[71] A. Tirachini and O. Cats, “COVID-19 and public transportation: Current assessment, prospects, and research needs,” *Journal of Public Transportation*, vol. 22, no. 1, 2020.

[72] C. Zakaria, A. Trivedi, E. Cecchet, M. Chee, P. Shenoy, and R. Balan, “Analyzing the impact of COVID-19 control policies on campus occupancy and mobility via passive WiFi sensing,” *arXiv preprint arXiv:2005.12050*, 2020.

[73] V. Osmani, I. Carreras, A. Matic, and P. Saar, “An analysis of distance estimation to detect proximity in social interactions,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 5, no. 3, pp. 297–306, 2014.

[74] W. J. Bradshaw, E. C. Alley, J. H. Huggins, A. L. Lloyd, and K. M. Esvelt, “Bidirectional contact tracing could dramatically improve COVID-19 control,” *Nature communications*, vol. 12, no. 1, pp. 1–9, 2021.

[75] S. Liu, Y. Jiang, and A. Striegel, “Face-to-face proximity estimationusing Bluetooth on smartphones,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 4, pp. 811–823, 2013.

[76] K. Katevas, H. Haddadi, L. Tokarchuk, and R. G. Clegg, “Detecting group formations using iBeacon technology,” in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, 2016, pp. 742–752.

[77] N. Palaghias, S. A. Hoseinitabatabaei, M. Nati, A. Gluhak, and K. Moessner, “Accurate detection of real-world social interactions with smartphones,” in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 579–585.

[78] D. J. Leith and S. Farrell, “Coronavirus contact tracing: Evaluating the potential of using Bluetooth received signal strength for proximity detection,” *ACM SIGCOMM Computer Communication Review*, vol. 50, no. 4, pp. 66–74, 2020.

[79] Z. Su, K. Pahlavan, and E. Agu, "Performance evaluation of COVID-19 proximity detection using Bluetooth le signal," *IEEE Access*, vol. 9, pp. 38 891–38 906, 2021.

[80] Google and A. 2020. (2020) Exposure notification: Bluetooth specification. [Online]. Available: https://blog.google/documents/70/Exposure_Notification_-_Bluetooth_Specification_v1.2.2.pdf

[81] P. Sapiezynski, A. Stopczynski, D. K. Wind, J. Leskovec, and S. Lehmann, "Inferring person-to-person proximity using WiFi signals," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 2, pp. 1–20, 2017.

[82] T. Altuwaiyan, M. Hadian, and X. Liang, "Epic: efficient privacy-preserving contact tracing for infection detection," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[83] S. Das, S. Chatterjee, S. Chakraborty, and B. Mitra, "An unsupervised model for detecting passively encountering groups from WiFi signals," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–7.

[84] K. Katevas, K. Hänsel, R. Clegg, I. Leontiadis, H. Haddadi, and L. Tokarchuk, "Finding Dory in the crowd: Detecting social interactions using multi-modal mobile sensing," in *Proceedings of the 1st Workshop on Machine Learning on Edge in Sensor Systems*, 2019, pp. 37–42.

[85] S. Shankar, R. Kanaparti, A. Chopra, R. Sukumaran, P. Patwa, M. Kang, A. Singh, K. P. McPherson, and R. Raskar, "Proximity sensing: Modeling and understanding noisy RSSI-BLE signals and other mobile sensor data for digital contact tracing," *arXiv preprint arXiv:2009.04991*, 2020.

[86] S. Jeong, S. Kuk, and H. Kim, "A smartphone magnetometer-based diagnostic test for automatic contact tracing in infectious disease epidemics," *IEEE Access*, vol. 7, pp. 20 734–20 747, 2019.

[87] S. Kuk, Y. Jeon, and H. Kim, "Detecting outdoor coexistence as a proxy of infectious contact through magnetometer traces," *Electronics Letters*, vol. 53, no. 19, pp. 1293–1294, 2017.

[88] A. Trivedi, C. Zakaria, R. Balan, A. Becker, G. Corey, and P. Shenoy, "WiFiTrace: Network-based contact tracing for infectious diseases using passive WiFi sensing," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 1, pp. 1–26, 2021.

[89] M. Dmitrienko, A. Singh, P. Erichsen, and R. Raskar, “Proximity inference with WiFi-colocation during the COVID-19 pandemic,” *arXiv preprint arXiv:2009.12699*, 2020.

[90] E. S. Lohan, J. Talvitie, P. F. e Silva, H. Nurminen, S. Ali-Löytty, and R. Piché, “Received signal strength models for WLAN and BLE-based indoor positioning in multi-floor buildings,” in *2015 International Conference on Localization and GNSS (ICL-GNSS).* IEEE, 2015, pp. 1–6.

[91] M. Cunche, A. Boutet, C. Castelluccia, C. Lauradoux, D. Le Métayer, and V. Roca, “On using Bluetooth-Low-Energy for contact tracing,” Ph.D. dissertation, Inria Grenoble Rhône-Alpes; INSA de Lyon, 2020.

[92] A. Montanari, S. Nawaz, C. Mascolo, and K. Sailer, “A study of Bluetooth Low Energy performance for human proximity detection in the workplace,” in *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom).* IEEE, 2017, pp. 90–99.

[93] S. Sareen, S. K. Sood, and S. K. Gupta, “IoT-based cloud framework to control Ebola virus outbreak,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 3, pp. 459–476, 2018.

[94] I. Nakamoto, S. Wang, Y. Guo, and W. Zhuang, “A QR Code–Based contact tracing framework for sustainable containment of COVID-19: Evaluation of an approach to assist the return to normal activity,” *JMIR mHealth and uHealth*, vol. 8, no. 9, p. e22321, 2020.

[95] D. J. Leith and S. Farrell, “Measurement-based evaluation of Google/Apple exposure notification API for proximity detection in a commuter bus,” *Plos one*, vol. 16, no. 4, p. e0250826, 2021.

[96] C. Gentner, D. Günther, and P. H. Kindt, “Identifying the BLE advertising channel for reliable distance estimation on smartphones,” *arXiv preprint arXiv:2006.09099*, 2020.

[97] J. Zimmerman, A. Tomasic, C. Garrod, D. Yoo, C. Hiruncharoenvate, R. Aziz, N. R. Thiruvengadam, Y. Huang, and A. Steinfeld, “Field trial of tiramisu: crowd-sourcing bus arrival times to spur co-design,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2011, pp. 1677–1686.

[98] “Public transit riders help one another avoid crowds with the Transit app,” 2020-11-02. [Online]. Available: https://www.masstransitmag.com/technology/passenger-info/mobile-applications/press-release/21160934/transit-app-public-transit-riders-help-one-another-avoid-crowds-with-the-transit-app

[99] N. Budhathoki and C. Haythornthwaite, “Motivation for open collaboration: Crowd and community models and the case of OpenStreetMap,” *American Behavioral Scientist*, vol. 57, no. 5, pp. 548–575,, 2015.

[100] A. Antrim, S. J. Barbeau *et al.*, “The many uses of GTFS data–opening the door to transit and multimodal applications,” *Location-Aware Information Systems Laboratory at the University of South Florida*, vol. 4, 2013.

[101] Q. Zervaas, “The definitive guide to GTFS-RealTime,” 2015.

[102] M. Ferreira, T. Dias, and J. Cunha, “Is Bluetooth Low Energy feasible for mobile ticketing in urban passenger transport?” *Transportation Research Interdisciplinary Perspectives*, vol. 5, 2020.

[103] ——, “Codesign of a mobile ticketing service solution based on BLE,” *Journal of Traffic and Logistics Engineering*, vol. 7, no. 1, 2019.

[104] J. Zhang, F. Zhang, C. Xu, Y. Wang, C. Tian, X. Li, B. Huang, and Z. Li, “A real-time passenger flow estimation and prediction method for urban bus transit systems,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, 2017.

[105] R. Meng, R. Grömling, and S. Nelakuditi, “Ridesense: Towards ticketless transportation,” in *2016 IEEE Vehicular Networking Conference (VNC*, 2016.

[106] M. Won, A. Mishra, and S. Son, “Hybridbaro: Mining driving routes using barometer sensor of smartphone,” *IEEE Sensors Journal*, vol. 17, no. 19, 2017.

[107] M. Oplenskedal, A. Taherkordi, and P. Herrmann, “Deepmatch: deep matching for in-vehicle presence detection in transportation,” in *14th ACM International Conference on Distributed and Event-based Systems*, 2020.

[108] K. Chintalapudi, A. Iyer, and V. Padmanabhan, “Indoor localization without the pain,” in *Sixteenth annual international conference on Mobile computing and networking*, 2010.

[109] J. Paek, J. Ko, and H. Shin, “A measurement study of BLE ibeacon and geometric adjustment scheme for indoor location-based mobile applications,” *Mobile Information Systems*, 2016.

[110] S. He and S. Chan, “Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 466–490,, 2015.

[111] S. Hemminki, P. Nurmi, and S. Tarkoma, “Accelerometer-based transportation mode detection on smartphones,” in *Proceedings of the 11th ACM conference on embedded networked sensor systems*, 2013, pp. 1–14.

[112] M. Kennedy, P. Spachos, and G. Taylor. (2019) BLE beacon indoor localization dataset. [Online]. Available: https://doi.org/10.5683/SP2/UTZTFT

[113] G. Mendoza-Silva, M. Matey-Sanz, J. Torres-Sospedra, and J. Huerta. (2019) BLE RSS measurements dataset for research on accurate indoor positioning. [Online]. Available: https://www.mdpi.com/2306-5729/4/1/12/htm

[114] P. Kasebzadeh, G. Hendeby, C. Fritsche, F. Gunnarsson, and F. Gustafsson, “IMU dataset for motion and device mode classification,” in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2017, pp. 1–8.

[115] H. Yan, Q. Shan, and Y. Furukawa, “RIDI: Robust IMU double integration,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 621–636.

[116] H. Yan, S. Herath, and Y. Furukawa, “RoNIN: Robust Neural Inertial Navigation in the wild: benchmark, evaluations, and new methods,” 2019, arXiv preprint arXiv:1905.12853.,.

[117] C. Chen, L. Zhao, C. X., W. Wang, A. Markham, and N. Trigoni, “Oxiod: The dataset for deep inertial odometry,” 2018, arXiv preprint arXiv:1809.07491,.

[118] P. Tiwari, V. P. Saxena, R. G. Mishra, and D. Bhavsar, “A survey of localization methods and techniques in wireless sensor networks,” *HCTL Open International Journal of Technology Innovations and Research (IJTIR)*, vol. 14, pp. 2321–1814, 2015.

[119] L. Cheng, C. Wu, Y. Zhang, H. Wu, M. Li, and C. Maple, “A survey of localization in wireless sensor network,” *International Journal of Distributed Sensor Networks*, vol. 8, no. 12, p. 962523, 2012.

[120] T. Janssen, M. Aernouts, R. Berkvens, and M. Weyn, “Outdoor fingerprinting localization using sigfox,” in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2018, pp. 1–6.

[121] S.-h. P. Won, W. W. Melek, and F. Golnaraghi, “A kalman/particle filter-based position and orientation estimation method using a position sensor/inertial measurement unit hybrid system,” *IEEE Transactions on Industrial Electronics*, vol. 57, no. 5, pp. 1787–1798, 2009.

[122] N. El-Sheimy, H. Hou, and X. Niu, “Analysis and modeling of inertial sensors using allan variance,” *IEEE Transactions on instrumentation and measurement*, vol. 57, no. 1, pp. 140–149, 2007.

[123] S. Roque-Cilia, E. I. Tamariz-Flores, R. Torrealba-Melendez, and D. H. Covarrubias-Rosales, “Transport tracking through communication in wdsn for smart cities,” *Measurement*, vol. 139, pp. 205–212, 2019.

[124] J. Zuo, S. Liu, H. Xia, and Y. Qiao, “Multi-phase fingerprint map based on interpolation for indoor localization using ibeacons,” *IEEE Sensors Journal*, vol. 18, no. 8, pp. 3351–3359, 2018.

[125] A. Coluccia and A. Fascista, “A review of advanced localization techniques for crowdsensing wireless sensor networks,” *Sensors*, vol. 19, no. 5, p. 988, 2019.

[126] S. P. Singh and S. Sharma, “Range free localization techniques in wireless sensor networks: A review,” *Procedia Computer Science*, vol. 57, pp. 7–16, 2015.

[127] M. Elbes, A. Alkhatib, A. Al-Fuqaha, and J. Qadir, “Using phase shift fingerprints and inertial measurements in support of precise localization in urban areas,” *Personal and Ubiquitous Computing*, pp. 1–12, 2019.

[128] 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*. ACM, 2000, pp. 32–43.

[129] K.-H. Lam, C.-C. Cheung, and W.-C. Lee, “New rssi-based lora localization algorithms for very noisy outdoor environment,” in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 2. IEEE, 2018, pp. 794–799.

[130] B. C. Fargas and M. N. Petersen, “Gps-free geolocation using lora in low-power wans,” in *2017 global internet of things summit (Giots)*. IEEE, 2017, pp. 1–6.

[131] D. F. Carvalho, A. Depari, P. Ferrari, A. Flammini, S. Rinaldi, and E. Sisinni, “On the feasibility of mobile sensing and tracking applications based on lpwan,” in *2018 IEEE Sensors Applications Symposium (SAS)*. IEEE, 2018, pp. 1–6.

[132] D. Niculescu and B. Nath, “Ad hoc positioning system (aps) using aoa,” in *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, vol. 3. Ieee, 2003, pp. 1734–1743.

[133] J. Xu, W. Liu, F. Lang, Y. Zhang, and C. Wang, "Distance measurement model based on rssi in wsn," *Wireless Sensor Network*, vol. 2, no. 08, p. 606, 2010.

[134] P. Nguyen-Hoang and R. Yeung, "What is paratransit worth?" *Transportation research part A: policy and practice*, vol. 44, no. 10, pp. 841–853, 2010.

[135] X. Zhao, Z. Xiao, A. Markham, N. Trigoni, and Y. Ren, "Does btle measure up against wifi? a comparison of indoor location performance," in *European Wireless 2014; 20th European Wireless Conference.* VDE, 2014, pp. 1–6.

[136] E. Miluzzo, M. Papandrea, N. D. Lane, H. Lu, and A. T. Campbell, "Pocket, bag, hand, etc.-automatically detecting phone context through discovery."

[137] Y. Kawahara, H. Kurasawa, and H. Morikawa, "Recognizing user context using mobile handsets with acceleration sensors," in *2007 IEEE International Conference on Portable Information Devices.* Ieee, 2007, pp. 1–5.

[138] J. Yang, E. Munguia-Tapia, and S. Gibbs, "Efficient in-pocket detection with mobile phones," in *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication.* ACM, 2013, pp. 31–34.

[139] F. Alonso-Martín, A. Castro-González, M. Malfaz, J. C. Castillo, and M. A. Salichs, "Identification and distance estimation of users and objects by means of electronic beacons in social robotics," *Expert Systems with Applications*, vol. 86, pp. 247–257, 2017.

[140] L. Maccari and V. Cagno, "Do we need a contact tracing app?" *Computer Communications*, vol. 166, pp. 9–18, 2021.

[141] G. Barthe, R. De Viti, P. Druschel, D. Garg, M. Gomez-Rodriguez, P. Ingo, M. Lentz, A. Mehta, and B. Schölkopf, "Pancast: Listening to Bluetooth beacons for epidemic risk mitigation," *arXiv preprint arXiv:2011.08069*, 2020.

[142] M. Al Qathrady, A. Helmy, and K. Almuzaini, "Infection tracing in smart hospitals," in *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob).* IEEE, 2016, pp. 1–8.

[143] R. Raskar, A. Singh, S. Zimmerman, and S. Kanaparti, "Adding location and global context to the Google/Apple exposure notification Bluetooth API," *arXiv preprint arXiv:2007.02317*, 2020.

[144] J. Bay, J. Kek, A. Tan, C. S. Hau, L. Yongquan, J. Tan, and T. A. Quy, “Bluetrace: A privacy-preserving protocol for community-driven contact tracing across borders,” *Government Technology Agency-Singapore, Tech. Rep*, 2020.

[145] P.-S. Loh, “Flipping the perspective in contact tracing,” *arXiv preprint arXiv:2010.03806*, 2020.

[146] A. Basiri, E. S. Lohan, T. Moore, A. Winstanley, P. Peltola, C. Hill, P. Amirian, and P. F. e Silva, “Indoor location based services challenges, requirements and usability of current solutions,” *Computer Science Review*, vol. 24, pp. 1–12, 2017.

[147] V. Shubina, A. Ometov, A. Basiri, and E. S. Lohan, “Effectiveness modelling of digital contact-tracing solutions for tackling the COVID-19 pandemic,” *The Journal of Navigation*, pp. 1–34, 2020.

[148] M. Girolami, F. Mavilia, and F. Delmastro, “Sensing social interactions through BLE beacons and commercial mobile devices,” *Pervasive and Mobile Computing*, vol. 67, p. 101198, 2020.

[149] N. . TC4TL. (2020) NIST pilot too close for too long (tc4tl) challenge evaluation plan. [Online]. Available: https://nist.gov/system/files/documents/2020/07/01/2020_NIST_Pilot_TC4TL_Challenge_Evaluation_Plan_v1p3.pdf

[150] V. Pasku, A. De Angelis, G. De Angelis, D. D. Arumugam, M. Dionigi, P. Carbone, A. Moschitta, and D. S. Ricketts, “Magnetic field-based positioning systems,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 2003–2017, 2017.

[151] A. Matic, V. Osmani, A. Maxhuni, and O. Mayora, “Multi-modal mobile sensing of social interactions,” in *2012 6th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops*. IEEE, 2012, pp. 105–114.

[152] M. L. . Laboratory. (2020) Structured contact tracing protocol. [Online]. Available: https://mitll.github.io/PACT/files/Structured%20Contact%20Tracing%20Protocol,%20V.%202.0%20(1.5).pdf

[153] S. Bian, B. Zhou, and P. Lukowicz, “Social distance monitor with a wearable magnetic field proximity sensor,” *Sensors*, vol. 20, no. 18, p. 5101, 2020.

[154] G. 2020. (2020) Exposure notifications BLE RSSI calibration procedure. [Online]. Available: https://developers.google.com/android/exposure-notifications/ble-attenuation-procedure

[155] P. Rajput, M. Chaturvedi, and V. Patel, "Opportunistic sensing based detection of crowdedness in public transport buses," *Pervasive and Mobile Computing*, vol. 68, p. 101246, 2020.

[156] J. Zimmerman, G. Tomasic, C. Hiruncharoenvate, R. Aziz, N. Thiruvengadam, Y. Huang, and A. Steinfeld, "Field trial of tiramisu: crowd-sourcing bus arrival times to spur co-design," in *SIGCHI Conference on Human Factors in Computing Systems*, 2011.

[157] G. H. Flores and R. Manduchi, "WeAllWalk: An annotated dataset of inertial sensor time series from blind walkers," *ACM Transactions on Accessible Computing (TACCESS)*, vol. 11, no. 1, pp. 1–28, 2018.

[158] C. Pluntke and B. Prabhakar, "Insinc: a platform for managing peak demand in public transit," *JOURNEYS, Land Transport Authority Academy of Singapore*, 2013.

[159] "Gamification as a way to induce behavioural change in mobility," 2019. [Online]. Available: https://civitas.eu/measure/gamification-way-induce-behavioural-change-mobility

[160] C. Pang, R. Pan, S. Wong, C. Neustaedter, and Y. Wu, "City explorer: Gamifying public transit trips while exploring the city," in *CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 2017.

[161] A. Marczewski, "Gamification: A simple introduction and a bit more," 2014.

[162] K. B, *Understanding gamification.* Chicago: ALA TechSource, 2015.

[163] "Contactless payment in public transport with Bluetooth Beacons," 2020. [Online]. Available: https://kontakt.io/blog/contactless-payment-public-transport-bluetooth-beacons/

[164] S. Maletz, "The rise of Bluetooth Beacons & public transit," 2017. [Online]. Available: https://medium.com/move-forward-blog/the-rise-of-bluetooth-beacons-public-transit-c43f3ea80b8d

[165] J. Lam, F. Mirzaei, and R. Manduchi, "Virtual tile routing for navigating complex transit hubs"," *Patent U.S. Application*, vol. 16, no. 932,301, 2021.

[166] F. Mirzaei, R. Manduchi, and S. Kurniawan, "Public transit accessibility: Blind passengers speak out," in *International Conference on Computers Helping People with Special Needs*, 2018.

[167] V. Trin and R. Manduchi, “Semantic interior mapology: A toolbox for indoor scene description from architectural floor plans,” in *Web3D 2019: The 24th International ACM Conference on 3D Web Technology*, 2019. [Online]. Available: https://escholarship.org/uc/item/2q07w48n

[168] F. Mirzaei, J. Lam, and R. Manduchi, “Accurate Self-Localization in transit stations: A case study,” in *EEE International Conference on Pervasive Computing and Communications Workshops*, 2020.

[169] V. Trinh and R. Manduchi, “Feeling your way around: Assessing the perceived utility of multi-scale indoor tactile maps,” in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020.

[170] M. Al-Razgan, S. Almoaiqel, N. Alrajhi, A. Alhumegani, A. Alshehri, B. Alnefaie, R. AlKhamiss, and S. Rushdi, “A systematic literature review on the usability of mobile applications for visually impaired users,” *PeerJ Computer Science*, vol. 7, p. e771, 2021.

[171] S. Herath, H. Yan, and Y. Furukawa, “Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, & new methods,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3146–3152.

[172] P. Ren, F. Elyasi, and R. Manduchi, “Smartphone-based inertial odometry for blind walkers,” *Sensors*, vol. 21, no. 12, p. 4033, 2021.