Localization, Beacon Placement and Mapping for Range-Based Indoor Localization Systems

Submitted in partial fulfillment of the requirements for
the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Niranjini Rajagopal
Master of Science, Electrical and Computer Engineering,
Carnegie Mellon University, Pittsburgh, USA
Bachelor of Technology, Electronics and Communications Engineering,
National Institute of Technology, Tiruchirappalli, India

Carnegie Mellon University
Pittsburgh, PA

August 2019

Abstract

Precise time and location are at the core of many Cyber Physical Systems (CPS) applications, as can be seen by the impact of the Global Positioning System. Indoor localization systems will enable applications ranging from navigation, asset tracking and secure device interaction to mixed reality experiences and emergency support (e911). However, indoor environments are full of barriers, which attenuate and scatter signals that make it challenging to provide high coverage in a cost-effective and reliable manner at scale. In this dissertation, we will show steps towards designing scalable indoor localization systems in a systematic manner that perform effectively across unpredictable environments and are compatible with several emerging technologies. Specifically, we focus on the class of range-based beacon technologies because (1) they can provide accurate ranges given line-of-sight, (2) they can instantly determine a location without requiring devices to move long distances and (3) there are growing standards for range-based technologies such as 802.11mc, BLE5 and ultra-wideband. Unfortunately, installing and mapping beacons is both expensive and time consuming, which continues to hinder adoption. Further, acquiring location from ranges in realistic settings can be inaccurate and slow when a low density of beacons and non-line-of-sight signals are encountered. While having an abundance of beacons increases accuracy and decreases the time to acquire the initial location, it also increases the cost of the system. This thesis aims to lower the barrier to adoption of range-based beacon technologies by reducing the infrastructure required while maintaining high performance in terms of accuracy and time taken to acquire the location and orientation. Our approach leverages additional sources of location information, like geometrical constraints from floor plans to acquire location with a reduced number of beacons, magnetic sensing to rapidly acquire orientation and visual inertial odometry for setup and continuous tracking. We present techniques for location estimation and system provisioning, including beacon placement and map generation, and show how these techniques apply to CPS applications like mobile augmented reality and first-responder localization.

Acknowledgments

I am deeply grateful to the numerous people who have had an impact on my PhD experience and this particular outcome, my dissertation.

Thanks to my advisors Anthony Rowe (chair of my thesis committee) and Bruno Sinopoli for guiding me through the PhD process, for giving me tremendous freedom in my research, and for being amazing mentors. I will never forget Anthony's always-open doors, and the numerous hours he spent with me in discussions and in giving feedback, especially in the early years when I was unfamiliar with both research, and the domain. Thanks to Anthony for pushing me to keep getting better and for taking great care in my growth over these years. His unending supply of energy and excitement kept my spirits high and helped me persist when the research hit dead ends and the demos didn't work. Thanks to Bruno for persistently encouraging me to embrace my own approach to research and for his confidence in my abilities. Bruno pushed me to be thorough and deep in my research pursuits, and guided me through the process of tackling messy real world systems problems in a systematic manner. When I started my PhD, I naively wanted to bridge theory and systems in the area of Cyber-Physical Systems without knowing what it takes to build such a bridge. I couldn't have asked for a better pair of advisors than Anthony and Bruno, for this journey. Most of all, I enjoyed my PhD, thanks to them.

Thanks to my committee members, Prabal Dutta and Brent Ledvina, for their mentorship over the years, for their confidence in my work and for their feedback. Thanks also to Brent for the opportunity to work with him at Apple through an internship. My interactions with Brent broadened my approach to research in sensing systems and made me think more critically about problems in localization. I thank Prabal also for his insightful questions, and for his advice on framing my work. I am amazed at how I left every discussion with Prabal with a new angle or perspective to my work.

Thanks to my fellow PhD students Patrick Lazik, Anh Luong and John Miller, and to master's students Sindhura Chayapathy and Krishna Kumar Reghu Kumar, whom I was lucky to mentor. Parts of this dissertation are the result of fruitful collaborations with these people over numerous hours spent together, in deployments, demos, experiments, implementations and brainstorming sessions. With them, I shared the excitement when after all the hard work, the location marker would show the true location. Several other colleagues have directly or indirectly influenced this work through collaborations and discussions. Thanks to Steven Aday, Adwait Dongare, Nuno Pereira, Oliver Shih, Nick Wilkerson, Xiaoqi Lin, Philip Petrakian, and Nikhil Choudhary.

I am thankful for several interactions and collaborations that I've stumbled into that helped me look at localization from a different perspective. In particular, I would like to thank Jie Gao and Haotian Wang for our recent collaboration on the theoretical approach to beacon placement, and to Srdjan Capkun and Mridula Singh for opening up a new angle for me to look at localization, from the security perspective. Thanks to my interactions with NIST, and my interactions at the COMPASS (Conference on Mobile Position Awareness Systems and Solutions) for helping me to take a step back and look at localization from a user's viewpoint. As a result, they made me think hard about which problems to focus my efforts on.

Thanks to my colleagues and to several fellow graduate students for fostering a fun and engaging environment and for the many technical and random discussions: to Khushboo Bhatia, Artur Balanuta, Max Buevich, Chris Palmer, Luis Pinto, John Costanzo, Mihovil Bartulovic, Xiaofei Liu, Sean Weerakkody, Paul Griffioen, Craig Hesling, Carmel Fiscko, Arvind Kandhalu, Matthew McCormack, Saurabh Shintre, Rajat Kateja, Soojin Moon, Mahmood Sharif, Sandeep Dsouza, Shreyas Venugopalan, Hoang Ngan Le and Omar Chowdhury.

Thanks to other faculty mentors I have had during my time in ECE at CMU: to Rohit Negi for inspiring me to be a good teacher, and to Pulkit Grover and Byron Yu for giving me an opportunity to develop my teaching. Thanks to the Chad Hershock, Heather Dwyer, Jessica Harrell, Emily Weiss, Katherine Walsh and my peer Graduate Teaching Fellows at the Eberly Center over the many years for enriching my experience at CMU.

Thanks to the staff at CyLab in the Department of Electrical and Computer Engineering, who have over the years helped me with the little and the big things to keep the research going: To Brigette Bernagozzi, Chelsea Mendenhall, Jamie Scanlon, Sheryl Benicky, Brittany Frost, Samantha Goldstein, Megan Kearns, Nathan Snizaski and Ivan Liang. A special thanks to Toni Fox, for her thoughtfulness and efforts to make our lab a better place in many ways.

Thanks to Emily Durham, without whom, the rest of this document would have commas and hyphens in the wrong places.

It is hard to imagine how my life during my PhD would have been, if not for my friends who were around. I thank them immensely. To Aishu, Shyama, Kaushik (Sivasubramanian) and Bargav, for being there in my earliest year at CMU and when I was new to the US; to Deeps, Divya (Sharma), Lavanya, James, Joy, Skb and Sudhanshu, Jyotsna, Gautam, Varun, Rajasekaran, Kaushik (Vaidyanathan), Harini, Toupie, Sahana, Krithika, Srinivas, Ji, Supreeth, Keshav, GC for all the good times; to Aranya, Utsav, Divya (Hariharan), Vikram, for all the good times and for being my guardians in Pittsburgh; to Vyas, Uday and Ashwati more than the good times, for bearing with me through the last year of my PhD; to Vivek for introducing me to most of these people.

Finally, thanks to my parents, Rajagopal and Madhuram, to my sister, Anjani, and to my husband, Harsha for their love, unwavering support and for believing in me more than I possibly can. In the spirit of range-beacon-based localization systems, they are my beacons. I dedicate this thesis to them.

I would like to acknowledge the various funding agencies that supported this research over the years. This research was funded in part by TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA; by CONIX Research Center, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA; by the NIST Public Safety Communications Research division; by NSF Building Innovation Capacity Award 1534114; by Bosch and Texas Instruments; and by the Samsung Ph.D. Fellowship and the Carnegie Mellon University William S. Dietrich II Presidential Ph.D. Fellowship.

Contents

List of Figures

List of Tables

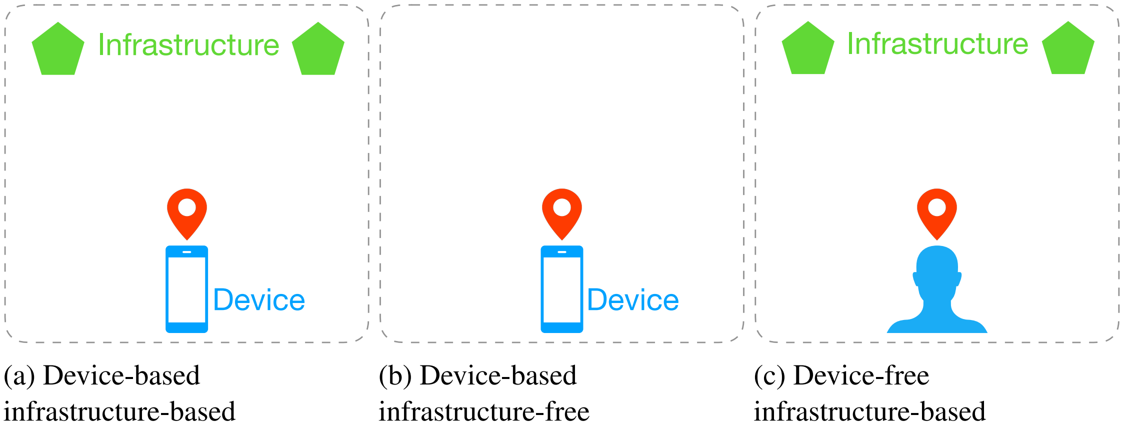
Chapter 1

Introduction

The ability to localize people and things indoors will impact diverse application domains ranging from search and rescue in disasters to next-generation augmented reality headsets. Indoor localization has been a long-standing challenge because no technology meets the requirements of being accurate, instant, compatible with existing infrastructure and inexpensive to install. Systems that produce accurate and instant estimates often require dedicated infrastructure to be set up and are not compatible with existing commodity devices like smartphones, tablets or headsets. Systems that currently do leverage existing infrastructure and devices are either not accurate or require an elaborate surveying process, which is expensive. Fundamentally, there is a trade-off between cost and performance across indoor localization solutions.

1.1 Indoor localization paradigms

We can better understand this trade-off by analyzing the localization paradigms that have emerged in the past three decades. For any indoor localization system, either the building environment or the target object to be localized requires some active or passive sensing capability. We refer to the instrumentation in buildings as *infrastructure* in short for *localization infrastructure*, and



(a) Device-based infrastructure-based

(b) Device-based infrastructure-free

(c) Device-free infrastructure-based

Figure 1.1: Indoor localization paradigms

refer to a target object with sensing capabilities as a *device*. Based on where the sensing capabilities lie, localization systems fall into three paradigms shown in Figure 1.1: device-based infrastructure-based, device-based infrastructure-free and device-free infrastructure-based. In addition to sensing, most localization systems need some prior information (that we refer to as *map*) to meaningfully convert the measured signals to location. To understand the performance-cost trade-off of a localization system, we should also consider the ease or difficulty of generating the maps required by the system.

Device-based and infrastructure-based indoor localization systems:
In this paradigm, infrastructure nodes are used as a reference and signaling between infrastructure nodes and a device is used to localize the device. Satellite-based positioning (e.g., GPS, GLONASS, Galileo) is an example of this paradigm in an outdoor setting. Indoors, the infrastructure can be either existing instrumentation in buildings (e.g., overhead LED lights, WiFi Access Points) or custom hardware deployed for localization. Examples of device-based infrastructure-based indoor localization include WiFi-based localization using signal strength [16, 100] or time-of-flight [133], ultrasonic or acoustic-based localization using time-of-flight or time-difference-of-arrival [65, 67], ultra-wideband-based localization using time-of-flight or time-difference-of-arrival [58, 59, 91], LED lights-based localization [63, 101, 102], and backscatter-based localization with RF [77, 84, 92]. These localization systems either require a precise map of the infrastructure nodes, or a map of the spatial variation of signals from the infrastructure. For instance, time-of-flight-based systems require the location of the infrastructure nodes, and signal strength-based systems either require the location of the infrastructure nodes along with a model of signal propagation or they require a prior map of the signal strength across different locations. The key advantage of these systems is that they can provide accurate localization given a high density of infrastructure. However, these systems are expensive due to infrastructure cost, as well as time and effort taken to set up, install and map environments. Further, a hindrance to adoption is often the lack of compatibility of these technologies with off-the-shelf commodity devices such as smartphones and tablets.

Device-based and infrastructure-free indoor localization systems:
In the second paradigm, a device localizes itself using on-board sensors such as camera, inertial measurement unit, magnetometer and microphone. Examples of this type of localization include systems that fuse inertial tracking with floor plan [141], systems that localize with the electromagnetic-field radiation from power lines [76] and most popularly, systems that localize using vision and depth sensors [32, 60, 73]. These systems are appealing since they do not require any additional infrastructure in the environment. However, they require maps of the measured signals across locations to compare against in order to localize. Generating these maps can be a labor-intensive process, and further, these maps have to be repeatedly updated if the environment changes. A single mode of sensing is often not sufficient to produce a unique location estimate, and these systems rely on mobility over time in order to capture enough data to uniquely localize.

Device-free and infrastructure-based indoor localization systems:
In the final paradigm, the object itself does not have sensing or control capability, but its presence impacts signals sensed by infrastructure nodes, which localize the object based on the sensed

signals. The infrastructure can be passive, for instance "outside-in" systems that localize using Kinect [46] or cameras [31]. In active systems, an RF transceiver might estimate an object's location based on changes in the wireless channel due to the presence or location of the object [12, 139, 144]. Recently, mmWave-based tracking and identification of persons [145] has been shown feasible in controlled environments. These systems are promising, since the object to be localized does not require any sensing and control capability. However, the RF-based methods are not yet robust to the unpredictabilities and dynamics of indoor environments and so far have been shown feasible in controlled lab environments, but not in realistic scenarios. These systems are also expensive due to cost of infrastructure, as well as the time and effort taken to set up, install and map environments. Compared to the infrastructure-based device-based localization systems, they require more resources in terms of the infrastructure and mapping. For instance, Amazon Go [4, 96] has shown accurate tracking of users in publicly accessible stores using cameras covering the ceiling.

1.2 Heterogenous opportunistic localization in the future

This wide variation in localization solutions is welcome given the variation in the application requirements, environment constraints, available sensors and semantics of indoor spaces. Unlike outdoors, where satellite-based positioning with inertial tracking is universally adopted for civilian applications, no single solution appears to solve all indoor localization problems. This dissertation advocates that in the future, multiple localization technologies will co-exist and devices will opportunistically localize based on the available infrastructure and sensors. This will require new tools and paradigms in order to bring structure to what is currently a haphazard and chaotic design process.

1.3 Range-based localization

In future opportunistic heterogenous localization ecosystems, infrastructure nodes will play a key role as they act as location references. In this dissertation, we focus on infrastructure-based device-based localization, with a focus on infrastructure nodes that are capable of ranging to target devices. We refer to these reference infrastructure nodes as *range-based beacons* and refer to these localization systems as *range-based localization systems*. Range-based localization systems estimate distance by measuring time of signal propagation between beacons and target devices using time-of-flight, time-difference-of-arrival or round-trip-time-of-flight.

1.3.1 Motivation

Range-based beacon systems are promising for several reasons. First, the accuracy expected from the range measurements is predictable under line-of-sight conditions. The accuracy is characterized by the signal bandwidth, the resolution of time stamping, clock-synchronization accuracy and other hardware factors [7, 108, 109], and is not environment or deployment-dependent. This

Localization technology	**Smartphone compatible?**	**Acquisition <1sec?**	**Accuracy <1m?**
WiFi	y	y/n	n
IMU	y	n	y/n
UWB (range-based)	n	y	y
Lidar (range-based)	n	y/n	y
BLE	y	y	n
Acoustic/ Ultrasonic (range-based)	y	y	y

Table 1.1: Suitability of current technologies for accurate location acquisition on smartphones

makes range-based localization systems predictable, and they have the potential to be accurate, if sufficient bandwidth and accurate time-stamping is available.

Second, range-based systems are capable of instant location acquisition. For example, if a device receives range measurements from a sufficient number of beacons at an instance of time, we can compute the location. In contrast, several other types of localization systems require the user to walk around until sufficient information is gathered from the sensors to compute the location. This is seen in Table 1.1, which compares suitability of various technologies for accurate location acquisition.

Third, range-based localization is applicable to a variety of physical layer technologies such as acoustic, UWB, BLE and WiFi, and hence encompass a broad class of devices and infrastructure. Ultrasonic beacons have been shown to provide accurate localization [66] on unmodified mobile devices. RF technologies are slowly finding their way into commodity devices. For instance, WiFi 802.11mc which supports round-trip-time-of-flight ranging [9] is currently implemented in Android Pie and is rolling out in some of the newer WiFi Access Points. Ultra-Wideband (UWB) ranging technologies have recently become popular for indoor localization due to readily available and highly accurate chipsets. Further, recent modifications to the physical layer are making UWB ranging secure with standards such as 802.15.4z [8, 10, 107]. Emerging modifications to the Bluetooth Low Energy (BLE) stack with support for time-of-flight ranging hold promise for prevalent peer-to-peer ranging among devices due to BLE's low power consumption and compatibility with commodity devices. Further, these emerging range-based technologies are reducing in price and power consumption, reducing the barrier to adoption.

1.3.2 Challenges

Though promising, in practice, range-based localization systems face a trade-off between cost and performance. This trade-off spans multiple layers of the system design process: beacon placement, mapping and localization. First, the system installer has to select the number of beacons to deploy. This is challenging, as a higher beacon density improves the localization performance but it also increases the system cost. The placement of beacons also impacts the localization accuracy, and there aren't systematic approaches to select beacon placements. Second, the system installer has to set up the beacons and create maps of the beacon locations, floor plan and any other spatially varying signatures used for localization. While on one hand, an accurate map is essential for good localization performance, on the other hand it is expensive (due to manual time, effort and the need for specialized surveying instruments) to accurately survey and map beacons and the environment. Further it is expensive to re-map the system regularly when beacons move or fail, or when new beacons are introduced. Finally, beacon-based systems face challenges in location estimation due to the nature of real-world indoor environments. In realistic environments, signals from line-of-sight beacons can be blocked and signals from non-line-of-sight beacons that reflect off walls are received. These result in either insufficient data to estimate location or inaccurate location estimates. To overcome this, the target device has to be moved around the building to gather more measurements, which increases the time-to-acquire location. In practice, beacons are over-deployed to improve acquisition time.

1.4 Contributions of this dissertation

This dissertation proposes a methodology for fusing additional sources of information to improve range-based indoor localization systems by reducing the infrastructure required while maintaining high performance in terms of accuracy and time-to-acquire the location and orientation.

This dissertation makes contributions to location and orientation solving, beacon placement and beacon mapping for range-based localization systems. This is represented in Figure 1.2 and elaborated below:

Beacon placement:
Challenge: Beacon placement in the real world faces the conflicting objectives of reducing the number of beacons (for cost reduction) and increasing system coverage, accuracy and resilience. Due to the lack of quantitative approaches to compare and evaluate beacon placements, current methods to deploy beacons either require domain experts who leverage intuition and heuristics, or over-provision indoor spaces with more beacons than required.
Contribution: We present a systematic approach to integrate the floor plan geometry and beacon coverage models to reduce the number of beacons while maintaining localization coverage. We define a metric to quantify the ability of a beacon placement to *Uniquely Localize* a floor plan. We build on this and introduce an indoor GDOP metric to quantify the localization accuracy of a beacon placement. We incorporate these metrics and design beacon placement algorithms [103] in an open source toolchain available to system designers.

Pedestrian-aided mapping:
Challenge: Range-based localization systems require a map as a reference to localize. The map includes the beacon locations and can include the floor plan as well as spatially varying vectored signals. Typically, the mapping is performed either by a robotic system or expensive surveying instrumentation, and the maps do not adapt to changing environments. This hinders the adoption of range-based technologies.
Contribution: We present a crowdsourced pedestrian-aided mapping process that simply requires users to walk around with phones that can be held in any orientation. We fuse visual inertial

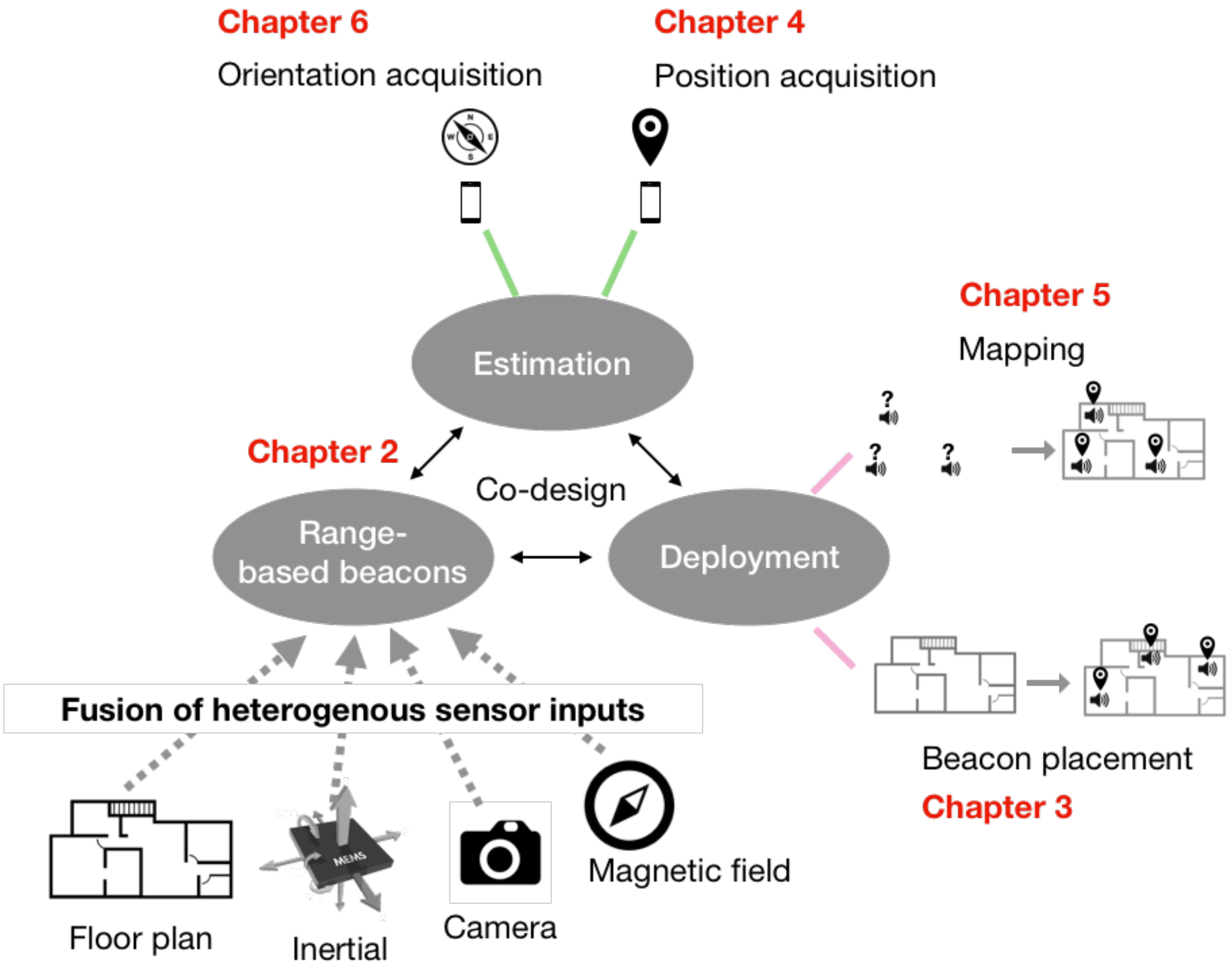


Figure 1.2: Overview of this dissertation. This dissertation proposes a methodology for fusing additional sources of information to improve range-based indoor localization systems by reducing the infrastructure required while maintaining high performance in terms of accuracy and time-to-acquire the location and orientation. This dissertation proposes solutions to the problems associated with deployment (beacon placement, mapping) and estimation (location acquisition and orientation acquisition).

odometry (VIO) on phones with beacon ranges to accurately track the phone's pose with respect to the beacons as users move around. The accurate tracking aids in mapping beacons, the floor plan and spatially varying signals in the environment.

Location acquisition:
Challenge: In realistic environments, LOS beacons are often blocked, and inaccurate range measurements are received from NLOS beacons. These under-defined scenarios result in inaccurate location estimates. Common solutions to cope with this either over-deploy beacons or rely on mobility and take time to converge on an estimate.
Contribution: We present a location solver that integrates the floor plan geometry and beacon coverage models to localize robustly in under-defined scenarios. The proposed location solver (1) considers feasible hypotheses of LOS and NLOS beacons among the received ranges and solves for locations under each hypothesis, (2) checks for consistency between the estimated location and the assumed hypothesis against the predetermined coverage information and (3) selects the most likely hypothesis-location pair. In addition, for most deployments, our approach guarantees that the localization accuracy does not degrade with the presence of any amount of NLOS measurements, as long as the expected LOS range measurements are received [104].

Orientation acquisition:
Challenge: Device orientation is important for applications such as wayfinding and mobile augmented reality, but range-beacon-based systems do not provide any orientation information. Orientation is acquired on mobile devices using visual features, but vision-based approaches often fail due to lack of texture, changes in lighting or changes in the location of objects in the environment, like moving furniture or people.
Contribution: We leverage accurate VIO trajectories to provide full vector magnetic field mapping that can be collected and used with devices placed in any orientation. We use the beacon ranges, VIO and magnetic field maps to acquire device orientation. This has the side effect of enabling multi-user (even cross-platform) applications that require users to be localized with respect to a common global reference without any sharing of visual feature maps [105].

1.5 Motivating applications

We motivate the approaches presented in this dissertation with two applications that require high-accuracy location as well as orientation, and have constrains on beacon placement and mapping: mobile augmented reality and firefighter localization.

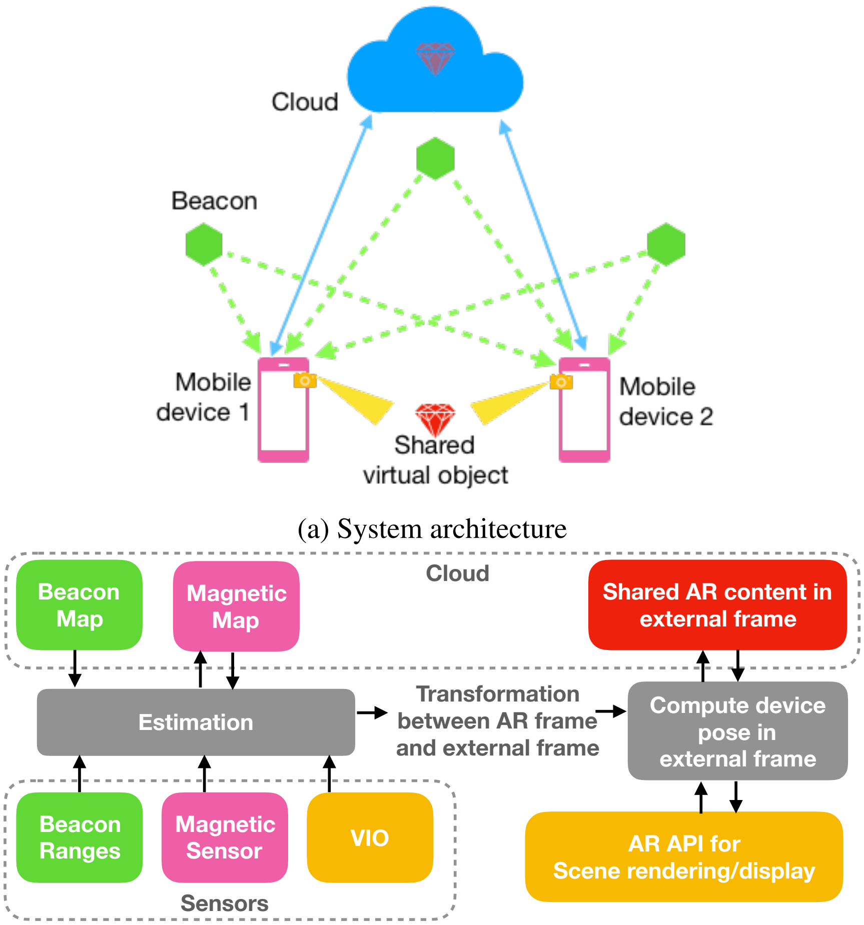
1.5.1 Application 1: Persistent mobile augmented reality

Recent advances in VIO and Simultaneous Localization and Mapping (SLAM) on devices ranging from headsets to smartphones have made AR an easily accessible commodity on an exciting number of platforms. Mobile AR APIs, like those found in Apple's ARKit and Android's ARCore, currently provide (1) VIO-based tracking, (2) scene understanding and (3) realistic rendering based on features and lighting from the environment. This can support applications like gaming and product visualization, where the user interacts with the virtual objects for a single session. For instance, the IKEA Place app allows users to experiment with different furniture layouts by rendering virtual furniture in their home. Advanced mobile AR and platforms such as Microsoft Hololens and Magic Leap are even able to support virtual content to *persist* across sessions (power cycles of the device) and across users (if the stored features are shared). Persistence of data on mobile devices opens up a variety of applications where users could now annotate items in large areas, like an office space with virtual signage and navigation instructions, or provide contextual control panels for things like projectors and automation equipment. Imagine adding live AR content to theaters, concert venues and sporting events where additional effects and information can be commonly broadcast to each viewer, even though they are at different vantage points in the space.

The key to seamlessly supporting persistent content between two AR sessions is that devices need to instantly and accurately relocalize themselves in six-degrees-of-freedom (6DOF) in a reference frame that is fixed and external to the devices. In state-of-art systems, this relocalization is achieved by comparing current camera data with stored visual features. While extremely promising, vision always suffers in environments devoid of visual features, with lack of texture, with changes in lighting and when the scenery changes over time due to moving furniture or people. As a result, in many cases, this relocalization process either takes an extended period of time or fails. In addition, visual relocalization in large areas requires searching through many candidate feature matches, which can become expensive if an initial location estimate is not provided. It is often the case that the user must walk around and view several areas of a scene before visual relocalization is able to take effect. We are optimistic that vision will continue to improve, but there are certain environments like office cubicles, hospitals or parts of airport terminals where even humans have trouble figuring out their location without exploring.

To overcome the limitations of a purely vision-based approach, we advocate using a localization infrastructure of range-based beacons which do not drift over time, are robust to environment dynamics and are based in a global reference frame that is not device- or session-dependent. Building on various approaches presented in this dissertation, we demonstrate multi-user persistent augmented reality on mobile devices by fusing data from beacon ranges, magnetic field and VIO to instantly and accurately relocalize mobile devices. Figure 1.3a illustrates the idea and Figure 1.3b shows the architecture for enhancing mobile AR [105]. Here, devices share persistent AR data through the cloud. This AR data is represented in an external fixed reference frame defined by the beacons. Each device reads/writes the AR data from/to the cloud and locally represents it in its own AR reference frame. Each AR session assumes a reference frame (AR frame) upon startup of an AR app, and the pose is locally tracked in this reference frame. In order to convert AR data between the AR frame and the external fixed frame, each device continuously estimates its transformation between its local AR frame and the external fixed reference frame and applies this transformation on the objects.

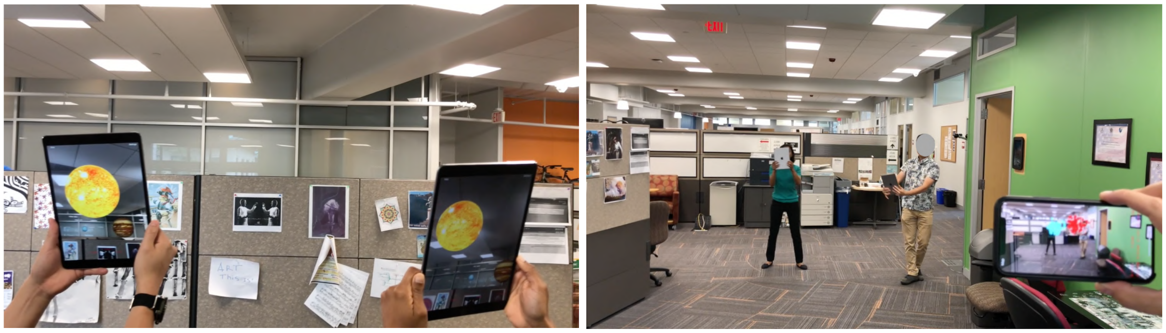
Figure 1.3b shows the process for estimating and applying this transformation, by building on various concepts introduced in this dissertation. We build and demonstrate this application with ultrasonic beacons described in Section 2.1. First, we set up beacons using the placement toolchain described in Section 3.6. The result of this process is the beacon map. As users walk around the environment, the magnetic field is mapped using the process described in Section 6.2,


Cloud
Beacon
Mobile device 1
Shared virtual object
Mobile device 2
Cloud
Beacon Map
Magnetic Map
Shared AR content in external frame
Estimation
Transformation between AR frame and external frame
Compute device pose in external frame
Beacon Ranges
Magnetic Sensor
VIO
Sensors
AR API for Scene rendering/display

(a) System architecture

(b) System components

Figure 1.3: Improving mobile augmented reality using range-based beacons



(a) AR annotation of spaces

(b) Find-a-friend application

Figure 1.4: Augmented reality demonstration applications

to generate the magnetic map by fusing beacon ranges, VIO and the magnetic field data. When users initiate the AR app, we incorporate the concepts from the floor-plan aware solver, described in Section 4.2, to initialize the location using beacon ranges. The orientation is acquired using VIO and the magnetic field. Subsequently, as users move around, the location and orientation are tracked by fusing VIO and beacon ranges. We estimate the transformation between the AR frame and the beacon frame using the process described in Section 6.2. This transformation is applied on the AR objects as they are transferred between the cloud and the AR API for scene rendering and display. The system continues to update the magnetic field map and the beacon map while updating the transformation in real-time.

Figure 6.11 demonstrates two persistent AR apps. In Figure 1.4a, two users are looking at a virtual solar system that is stationary in the environment. This app is representative of applications where indoor environments are annotated with AR content tied to physical locations. Figure 1.4b shows a user seeing two other users in AR in real-time. The user's device shows the location of the other users in red and cyan color augmented points. This is representative of a find-a-friend application. These apps present a proof-of-concept for using beacons integrated with VIO to provide persistent AR on mobile devices.

In the future, our approach could easily be integrated with low-level visual relocalization to support areas with sparse or no beacon coverage. Given a high-confidence location and orientation, the search space for visual relocalization can be dramatically reduced and the tolerance for matches can be relaxed. Though demonstrated on a mobile AR platform, this same approach would easily apply to headsets or similar localization platforms that require full pose information.

1.5.2 Application 2: An infrastructure-free localization for firefighters

Tracking the location of firefighters and search and rescue teams is critical for both safety and efficiency. Currently, firefighters often rely on either rope search or sectored sweep coverage strategies that can be both dangerous and slow. Given the hostile nature of burning structures and the time-critical nature of missions, *we require a system that can track firefighters without any pre-installed internal and limited external infrastructure, and without assuming knowledge of the structures layout.* For a system to be practically adopted at scale, it also needs to be *low-cost and extremely simple to configure and deploy.*

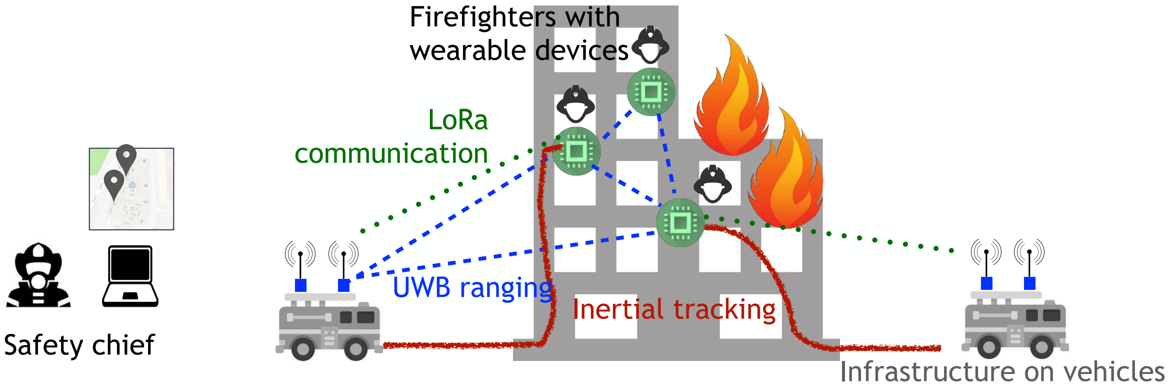


Figure 1.5: Architecture for infrastructure-free firefighter localization

To solve this problem, we propose an approach that combines time-of-flight Ultra-Wide Band (UWB) ranging radios and inertial measurement units. The system architecture is shown in Figure 1.5. We envision a system consisting of a small number of external beacons present on emergency service vehicles that drive up to the building perimeter and a small wearable unit attached to each firefighter's belt or air pack. Each external beacon would have a GPS receiver, sub-GHz LoRa radio, UWB radio and air pressure sensors. The wearable unit on each firefighter would have similar hardware with the addition of an IMU.

Next, we describe how the concepts introduced in this dissertation apply to the infrastructure-free localization system. Since we cannot rely on pre-existing infrastructure in the building, the beacons on the emergency vehicles act as a reference for localization. As firefighters exit the vehicles and move towards the buildings, we begin to perform simultaneous localization of the firefighters and mapping of the beacons by applying the concepts that we introduce in Section 5.4. To aid in quicker convergence of the orientation, we use the magnetic field information and rely on the compass outdoors to acquire a coarse orientation upon startup, as in Section 6.2. As firefighters move into the building, we continue to track their location by combining the UWB ranges and inertial sensor data by building on the concepts in Section 5.2. Over time, by combining the inter-device ranges and the mobility data from sensors, the joint location of firefighters over time form a virtual dense network; the location of the nodes of the network can be solved by graph realization techniques. In the prototype system that we built, we (1) localize the mobile devices by ranging to beacons, (2) relay this information in real-time to an external console, representative of the safety chief's display, (3) allow the safety chief to provide inputs through a GUI on the location of hazards and (4) use augmented reality to overlay the hazard information and the location of other mobile devices on each mobile device's display. Figure 1.6 shows the snapshot of our prototype system. The display on the left shows the safety chief's interface. The red and blue dots are tracking the real-time position of two mobile devices. The safety chief places fire hazards (shown in orange triangle), based on information conveyed by a part of the team. This information is relayed in real-time to the mobile device (shown on the right) and displayed in augmented reality to the rest of the team. In the future, we can build on concepts in mobile network localization [115] and also leverage drones to expand the localization coverage. We also envision that firefighters' self-contained breathing apparatus (SCBA) face

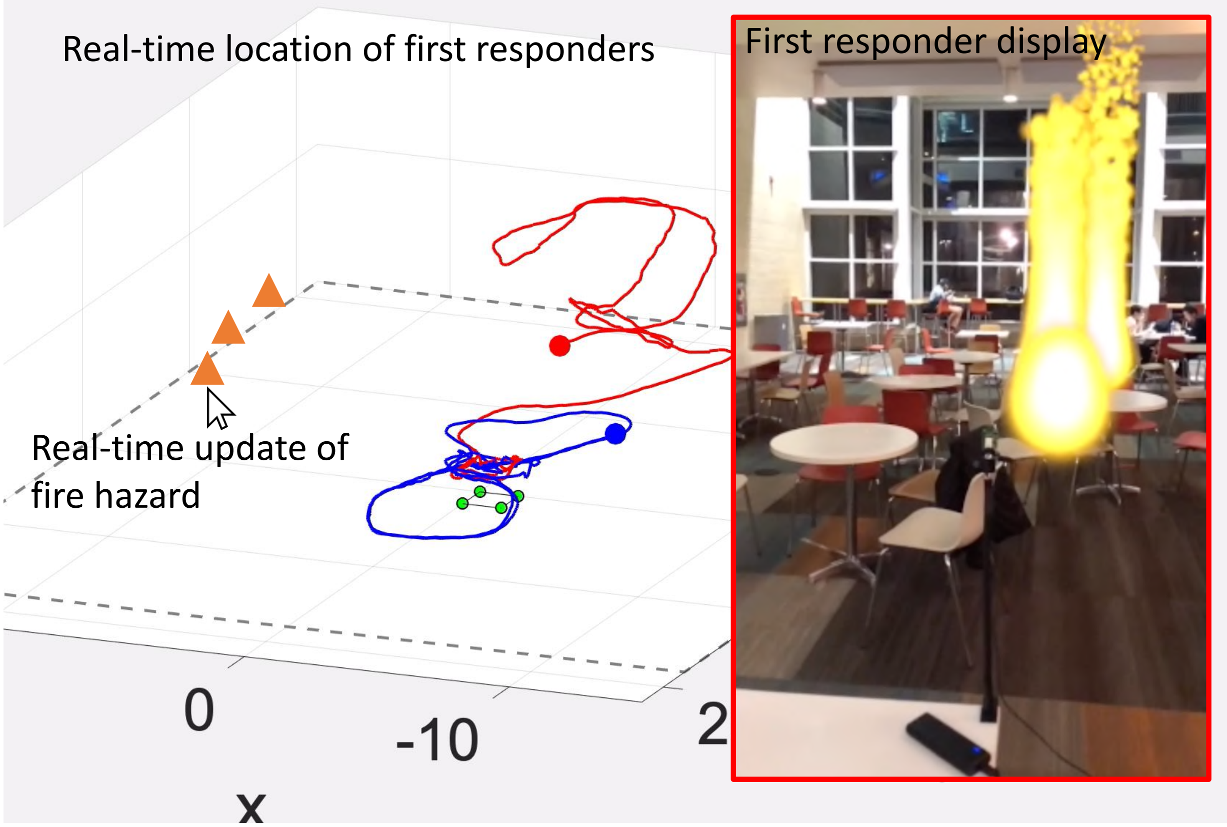


Figure 1.6: Snapshot of the prototype system for infrastructure-free firefighter localization system

masks can be provisioned with a smart display that allows visual information such as the location of other firefighters and the presence of hazards to be overlaid on the display. Being able to precisely estimate location and orientation of all devices instantly in the same frame of reference is key to enabling this application.

1.6 Organization of this dissertation

This dissertation is organized as follows. In Chapter 2, we describe two ranging platforms on which we implement and evaluate our approaches. Then we ask the question, “Where should one place the beacons?” This question is answered in Chapter 3, and we provide a toolchain that system designers can use to experiment with beacon placement strategies. Then we answer the question, “How should one estimate the location accurately and instantly?” In Chapter 4, we present a solver for location acquisition. In Chapter 3 and Chapter 4, we do not consider user mobility. In Chapter 5 and Chapter 6, we consider user mobility. In Chapter 5, we use mobility to create a map of the floor plan and the beacons. We build on this in Chapter 6 to create a map of spatially varying signals, in particular the magnetic field, which we then use for orientation acquisition. We conclude with future directions in Chapter 7.

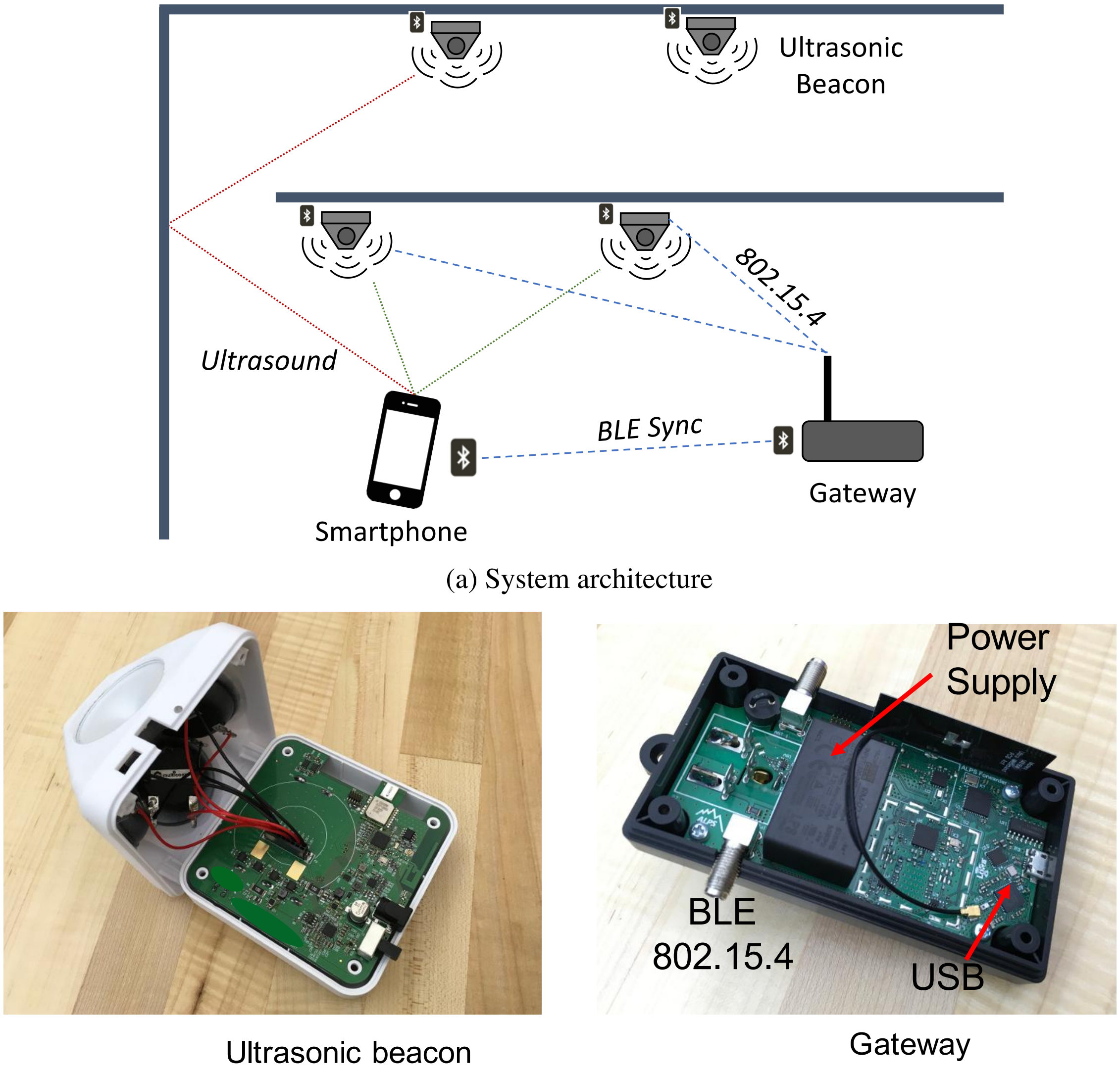
Chapter 2

Range-Based Beacon Platforms

The methods presented in this dissertation are implemented and evaluated against two ranging platforms: an ultrasonic ranging platform that we designed and an ultra-wideband (UWB) ranging platform that is commercially available. The ultrasonic ranging platform localizes unmodified phones and the UWB platform localizes other UWB tags.

2.1 Ultrasonic ranging platform

The ultrasonic platform performs ranging to unmodified phones and is elaborated on in [65, 66, 67, 104]. Figure 2.1 shows the system architecture and the hardware design of our beacons and gateway board. The platform, shown on the left, is based on a multi-standard BLE and IEEE 802.15.4 SoC (TI CC2650) connected to a $192kHz$ audio codec (running at $48kHz$), a MEMS microphone (for inter-beacon ranging or for future uses like ultrasonic motion detection) and an array of speakers connected to two Class D speaker amplifiers. The hardware can transmit two arbitrary sound waveforms up to $80kHz$ from one to four speakers simultaneously. The gateway board, shown on the right, contains similar hardware with the addition of a IEEE 802.15.4/BLE power amplifier and an FTDI USB-to-serial interface for connecting with a computer. The beacon nodes are synchronized using IEEE 802.15.4 from the gateway devices and then broadcast BLE packets that can be used to trigger normal Bluetooth proximity services on mobile phones, as shown in Figure 2.1a. These wakeup signals can in turn begin decoding ultrasound for improved accuracy. We adapt the signal and demodulation approach from [66] that utilized ultrasonic chirps that are just above human hearing range, but can still be detected by mobile devices like smartphones, tablets and computers. Since upchirps and downchirps are primarily orthogonal, we can transmit both at the same time without interference. Each chirp in our system is sized to be $110ms$ in length with $2.6ms$ between successive chirps. To support four sectors, we can shift one pair of up/down transmissions in time with a neighboring beacon. Each beacon is given a unique time slot that is $350ms$ in length to transmit its ranging signal. It is also worth noting that the overlapping transmissions are not only orthogonally coded, but also transmitted in opposite directions in the horizontal plane.



(a) System architecture

(b) Hardware

Figure 2.1: Ultrasonic ranging: System architecture and platform design

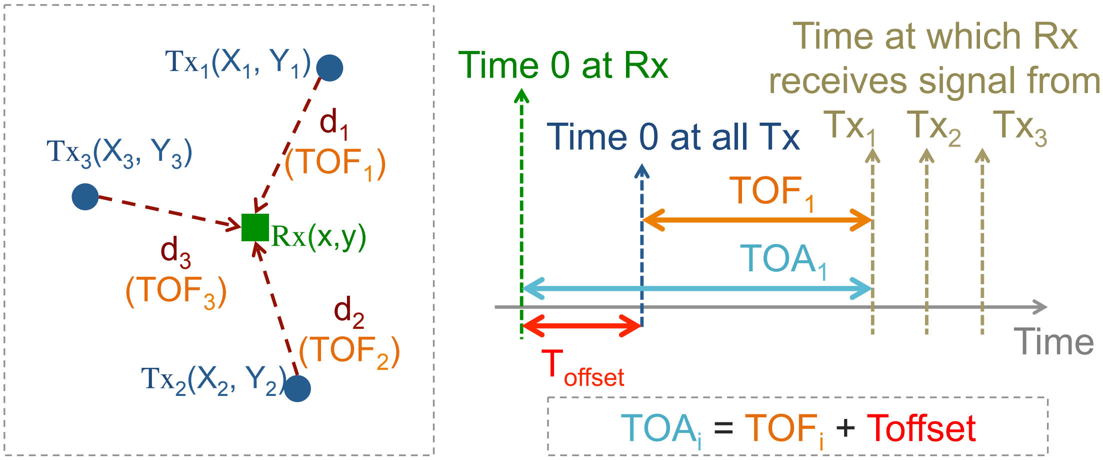


Figure 2.2: Achieving ToF ranging from TDoA by estimation of timing offset between target device and beacons

2.1.1 Time synchronization

As seen in Figure 2.1a, the beacons are synchronized using 802.15.4 packets. However, the phones are not capable of being synchronized with 802.15.4. With the beacons synchronized to each other but not to the phone, we can perform localization using TDoA. However, if we recover the clock from the network of beacon on the phone, we can perform ToF ranging which requires one less beacon than TDoA. As is often the case with range-based systems, time synchronization between beacons and the smartphone is challenging due to the timing uncertainties in the smartphone. To overcome this challenge and synchronize the phone to the beacons, we propose two approaches below that allow us to recover the global clock on the device being tracked.

Approach 1: TDoA to ToF

In an early version of the system, we synchronized the phone to beacons by performing TDoA localization when sufficient beacons were in LOS, then estimated the timing offset between the beacons and phone, and used this timing offset to synchronize the phone to the beacons to subsequently perform ToF ranging. This process is described below:

Figure 2.2 shows the layout of three transmitters and a receiver in 2D space, and their corresponding notions of time. We consider the receiver's clock to be offset by T_{offset} from the transmitter's clock. Synchronization is achieved by estimating this offset. Typically this time offset is not estimated, since the TDoA equations are used to directly estimate the position of the receiver [56]. However, the time offset can be obtained easily once the position has been estimated, as explained below.

(X_i, Y_i) denotes the position of transmitter i for $i = 1, 2, 3$ and is assumed to be known. The position of the receiver (x, y) is unknown. d_i is the distance between transmitter i and the receiver and is given by:

$$d_i(x, y) = \sqrt{(X_i - x)^2 + (Y_i - y)^2}$$

The ToF of the audio signal from transmitter i to the receiver is given by:

$$TOF_i = \frac{d_i(x,y)}{V}$$

where V is the speed of sound.
The corresponding arrival time of the signal measured by the receiver is the TOA_i, given by:

$$TOA_i = TOF_i + T_{offset}$$
$$TOA_i = \frac{d_i(x,y)}{V} + T_{offset}$$

The receiver needs to estimate T_{offset} given TOA_i and (X_i, Y_i) for $i = 1, 2, 3$. To estimate the T_{offset}, we first estimate the position of the receiver. To estimate (x, y), we use the standard multilateration technique [56] by eliminating T_{offset} and arrive at the TDoA equations. We then find the (x, y) that minimizes the sum of squares of error (ξ) in TDoA.

$$\begin{aligned} Measured\, TDOA_{ij} =& TOA_i - TOA_j \\ True\ TDOA_{ij} =& \frac{d_i(x,y) - d_j(x,y)}{V} \\ \xi_{TDOA_{ij}}(x,y) =& [TOA_i - TOA_j - \frac{d_i(x,y) - d_j(x,y)}{V}]^2 \\ (\hat{x}, \hat{y}) =& \operatorname*{argmin}_{x,y} \sum_{\substack{(i,j) \\ 1 \le j \le N \\ j \ne i}} \xi_{TDOA_{ij}}(x,y) \end{aligned}$$

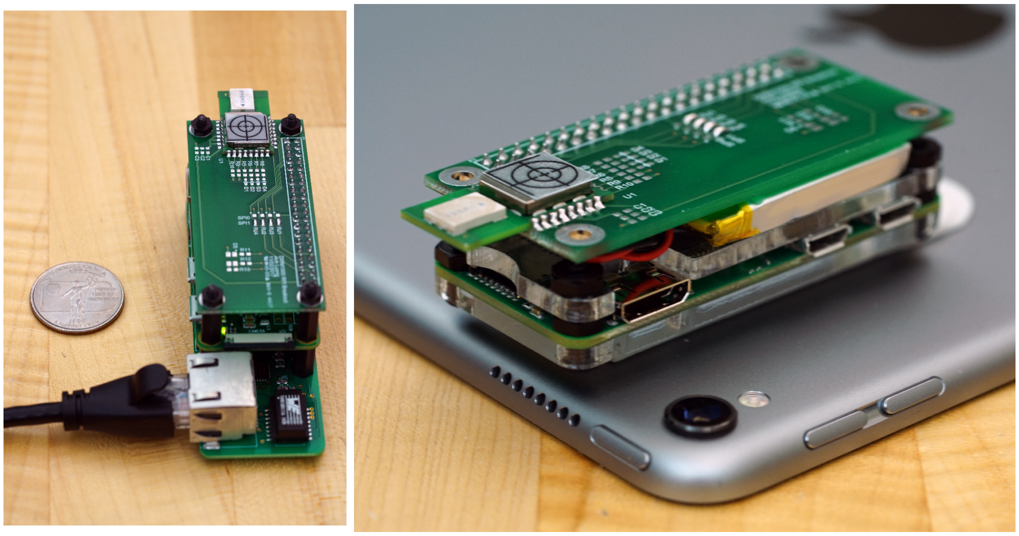
We next estimate T_{offset} from $(\hat{x}, \hat{y})$ and the TOA by:

$$T_{of\hat{f}set} = \frac{1}{3}\Bigg(\sum_{i=1}^{3}\Big(TOA_i - \frac{d_i(\hat{x},\hat{y})}{V}\Big)\Bigg)$$

In this manner, as part of the TDoA calculation, it is possible to estimate the instant when each signal was originally transmitted, and we can use this to synchronize the audio stream with respect to global time. This can then be used as a reference for application-level time-stamping. Time-stamping of audio events based on their position in a buffer completely removes any sources of delay from the operating system or networking stack. Given the relatively small amount of jitter seen when sampling audio, it also stands as a reasonable alternative for synchronizing other events, e.g., to perform cooperative ranging between two mobile phones.

Approach 2: BLE-based time synchronization

In the second approach, we do not rely on TDoA-based positioning at startup. Instead, the system supports ToF upon startup. We use controlled BLE advertisement packet arrival timing to tightly synchronize between an application running on a smartphone and beacons. We design a software Phase Loop Lock (PLL) that is able to recover the clock from our beacons after just



(a) UWB beacon

(b) UWB tag with tablet

Figure 2.3: UWB platform based on Decawave used for evaluation

a few cycles [104]. Using this approach, the resulting error during the steady state of our clock synchronization scheme is well under $1ms$, and 96% is within $[-200, 200]\mu$s. According to data from [67], the clocks on iPhones are stable enough that they can remain synchronized to below one millisecond for tens of minutes. We use this platform for evaluating the proposed algorithms and for demonstrating the multi-user persistent mobile augmented reality system and applications.

We use this platform for the models used in beacon placement in Chapter 3, for evaluating the location acquisition in Chapter 4, for evaluating the floor plan mapping in Chapter 5 and for building a prototype of the multi-user persistent augmented reality applications presented earlier in Chapter 1.

2.2 Ultra-wideband (UWB) platform

We built a prototype ToF RF ranging system using Decawave UWB radios. In contrast to acoustic ranging, RF signals penetrate surfaces more easily, representing another common signal propagation model. Decawave currently has a small UWB and BLE module [6] that can be easily attached to phones in order to range with UWB and provide position information over Bluetooth. In our prototype, we use a DWM1000 module connected to a Raspberry Pi Zero W shown in Figure 2.3a. The UWB hardware is identical for beacons and tags. For the mobile receiver, we stuck a UWB tag to a smartphone/tablet shown in Figure 2.3b. In our implementation, we used three-way ranging between beacons and tags. For debugging and simplicity, the beacon sensor data and the VIO data from the mobile devices are published to a MQTT topic. The solver subscribes to this topic and estimates the location.

We use this platform for evaluating the beacon mapping in Chapter 5, the pose acquisition in Chapter 6 and for building a prototype of the infrastructure-free localization system for firefighters presented earlier in Chapter 1.

Chapter 3

Beacon Placement

In this chapter, we answer the question - For a range-based localization system, given the floor plan, where should the beacons be placed? [103]

3.1 Introduction

Beacon placement in the real world faces the conflicting objectives of reducing the number of beacons (for cost reduction) and increasing system coverage, accuracy and resilience. Current methods to deploy beacons in order to ensure comprehensive localization coverage either require domain experts who leverage intuition and heuristics, or over-provision indoor spaces with more beacons than required. Minimizing the number of beacons is particularly important as these systems transition from small deployments, mainly used for demonstration purposes, to commercial ones across large real-world spaces such as airports, museums, malls and industrial buildings. In these environments, a systematic beacon placement methodology will have a significant impact in terms of cost savings, thus facilitating adoption of emerging range-based localization technologies. In this chapter, we aim to examine the beacon placement problem, with a focus on practical implications while optimizing for coverage and beacon count.

First, we attempt to reduce the number of beacons. Existing approaches for trilateration require three or more beacons (in 2D) to determine a unique location solution. However, it becomes expensive to provision all regions in a building with three or more beacons. To reduce the beacons, our key insight is that it is possible to compute a unique location estimate without ambiguity with only two beacons by leveraging prior knowledge of the map and a model of the beacon coverage. To translate this idea to a beacon placement strategy, first we describe a way to compute whether a location is Uniquely Localizable by a beacon configuration. Then we define a new metric that quantifies the beacon configuration quality based on the Unique Localization coverage it provides for a floor plan. With this basis, we design a beacon placement strategy that aims to minimize the number of beacons required, while maximizing the localization coverage.

Next, we quantify the localization accuracy provided by a particular beacon placement. For this, we adopt the Geometric Dilution of Precision (GDOP) metric from the GPS literature. GDOP, a unitless quantity, is an analytical function of the geometry between the beacons and the target. It bounds the location accuracy attainable based on the geometry of the satellites [81].

While we could adopt the GDOP metric from outdoors to indoors, there are differences due to the symmetric deployment of satellites around the earth and the rather asymmetric deployment required of beacons indoors due to barriers and signal attenuation through walls. As a result, in low beacon density areas indoors, multiple ambiguous solutions produce errors, which are unfortunately not reflected by the GDOP. Further, under certain poor beacon geometries, the standard GDOP metric can grow towards infinity, making it difficult to incorporate it to quantitatively compare beacon configurations. To overcome these problems, we define a modified indoor GDOP metric that first incorporates whether a location is Uniquely Localizable and then computes the Cumulative Distribution Function of the indoor GDOP over the regions that are localizable. This directly allows us to compare the localization quality across beacon configurations. The localization quality is an indication of the beacon configuration quality. This guides us to design a beacon placement strategy that aims to minimize the number of beacons required, while maximizing the localization accuracy in addition to coverage.

We then integrate these metrics and insights into a beacon placement algorithm. The algorithm uses the coverage of candidate beacon locations to segment the floor plan into disjoint zones, each of which is covered by a unique candidate set. Then, it iteratively selects a beacon among the ones that cover the largest unlocalized zone. The selection criteria is greedy; the beacon that maximizes the improvement in the beacon configuration quality is selected. The beacon configuration quality can be specified by the system designer to be based on either coverage or accuracy.

We also have to consider several practical factors while solving this problem. First, beacon coverage depends on the physical technologies, which vary in their maximum range and signal permeability through walls. For instance, acoustic/ultrasonic signals are confined to walls, while RF signals exhibit high penetration. Second, physical factors constrain the deployment. For instance, one may prefer to place beacons with convenient access to power outlets or where they do not disrupt the aesthetics of the space. Third, indoor spaces have rich semantics that lead to different localization accuracy requirement across different areas. For instance, room-level accuracy might be sufficient in certain areas, while sub-meter accuracy within a room might be required for audio guides in museums. To accommodate these practical constraints, we designed a MATLAB-based toolchain where users can provide a floor plan and specify the beacon range and potential locations to place beacons. We have implemented the beacon placement algorithms in the toolchain. The toolchain is open source and available at https://github.com/WiseLabCMU/BeaconPlacementToolchain.git.

This chapter is organized as follows. In Section 3.2, we describe the related work in beacon placement. In Section 3.3, we formulate the beacon placement problem and describe the model and our assumptions. In Section 3.4, we propose a method to integrate the floor plan information with beacon coverage. We introduce our approach of localizing with two beacons and the concept of unique localizability in Section 3.4.2. In Section 3.5, we build on this to design an indoor-GDOP metric that accounts for localization accuracy in addition to coverage. We incorporate these metrics to design a beacon placement algorithm that we describe in Section 3.6. Finally, in Section 3.7, we illustrate the MATLAB-based toolchain which allows users to specify deployment inputs and suggests beacon locations based on the proposed algorithms. We present the results of the beacon placement approaches in Section 3.8.

3.2 Related work

In this section, we first discuss prior work in beacon placement that ensures coverage across all regions in a building, and then discuss prior work in beacon placement that accounts for localization accuracy.

Beacon placement for coverage

Mathematically, the problem of beacon selection for unique localization is closely related to the classical Art Gallery problem in computational geometry [90]. The goal of the Art Gallery problem is to find the minimal set of guards such that every region in a floor plan is covered by at least one guard. The Art Gallery problem is NP-hard even for simple polygons [69]. For any simple polygon P with n vertices, it has been proven that $\lfloor n/3 \rfloor$ guards are always sufficient [27] and sometimes necessary. For a polygon P with h holes, it was shown that P can be guarded with $\lceil \frac{n+h}{3} \rceil$ guards [20, 50]. This problem has been proved to be APX-hard [36], implying that it is unlikely that any approximation ratio better than some fixed constant can be found. In [43], it is shown that a logarithmic approximation may be achieved by discretizing the input polygon into convex subregions. Also related to our work is the *k-coverage* set problem where the goal is to find the minimum set that covers all the points at least k times. The k-coverage problem is NP-hard as well, but several approximation algorithms have been proposed for this class of problems. Cormen [30] applied a greedy approach for the *k-coverage* problem with a $O(k \log n)$-approximation solution, where n is the number of points. The ε-net technique was introduced for this problem when the sets have constant VC-dimensions and was shown to achieve approximation factor $O(\log \text{OPT})$, where OPT is the optimal solution.

Our problem is not the Art Gallery problem or the *k-coverage* problem, because each point in the domain has to be Uniquely Localized, which can be achieved by two or three beacons, depending on geometry.

Beacon placement for localization

In order to find beacon configurations that minimize the localization error at a given target location, several authors have studied optimal beacon geometries that minimize the Cramér-Rao bound (CRB) [57] or the GDOP [94, 121]. For $N \geq 3$ beacons, optimal placement can be obtained when the adjacent beacons subtend an angle of $\frac{2\pi}{N}$ or $\frac{\pi}{N}$ about the target, and for $N = 2$ beacons, optimal placement is when the two beacons subtend an angle of $\frac{\pi}{2}$ about the target [18, 113].

Though the optimal placement for a single target is well understood, the optimal placement for multiple target locations, a target trace and a target area for range-only systems are still open problems. [128] shows that the generalized sensor placement problem is at least as hard as the k-center problem, which is NP-complete. They present a solution based on integer linear programming for a triangulation-based system, but the complexity for trilateration is similar. Due to the complexity of the problem, most proposed solutions involve designing heuristics and utilizing optimization techniques. [13] explores computational-geometry-based heuristics for determining the location of beacons given a predefined trace of a robot by defining a utility metric that is a function of the number of beacons in range and the convex hull of beacons. The quality of a configuration is evaluated using the integral of the utility function. Our work is similar in the respect that we also account for the coverage of the beacons, but differs since the metric they propose is purely based on coverage and not accuracy. [142] proposes a heuristic for *Quality of Trilateration* that is based on the probability that a location estimate is within a given radius of the true location. However, this approach is for a network of nodes where a node localizes itself with respect to three other nodes given a prior distribution on the expected inter-node distances. [70] and [110] propose beacon placement algorithms based on their own metrics to quantify the quality of a beacon placement, which is a function of the GDOP over the desired localization area and ratio of the area which cannot be localized. [70] classifies areas that are localizable based on whether the GDOP is above or below a threshold and use a genetic algorithm approach for placement. [110] considers the average GDOP over the areas that are localizable and implements a meta-heuristic optimization strategy. [62] looks at the problem of beacon placement while localizing with range-based beacons with a limited field of view. They propose a heuristic based on the GDOP being below a certain threshold and propose a placement algorithm for the same, but they consider only a single room case wherein the beacons are placed on the periphery of the room, avoiding any ambiguity in solution. These approaches are similar to our work, but do not consider the minimal beacon count placement for areas and require three or more beacons to provide coverage.

3.3 Problem formulation

The beacon placement problem is as follows: Given a floor plan and a set of candidate beacons, our goal is to select the smallest beacon configuration that can localize all points in a floor plan.

3.3.1 Definitions

We define the following terms:

Definition 3.1. (Floor plan X): The floor plan represents the regions or points to be localized. We represent the floor plan as a polygon. Any permanent walls and obstructions inside are modeled as holes in the polygon. We discretize the polygon to a set of points x_i, where x_i represents the (x, y) coordinates of the i^{th} point.

$$X = \{x_i | 1 \leq i \leq N_X\} \tag{3.1}$$

Definition 3.2. (Candidate beacons P): The set of locations where we can place a beacon are called the candidate beacons. We represent this set as P. Each element p_i represents the (x, y) coordinates of the candidate location. The beacons are placed inside the floor plan.

$$P = \{p_i | 1 \leq i \leq N_P, p_i \in X, N_P \leq N_X\} \tag{3.2}$$

$$P \subseteq X \tag{3.3}$$

Definition 3.3. (Beacon configuration B): Any subset of candidate beacons form a beacon configuration B.

$$B = \{b_i, | 1 \leq i \leq N_B, N_B \leq N_P\} \tag{3.4}$$

$$B \subseteq P \tag{3.5}$$

$$\tag{3.6}$$

The problem of localizing all regions is solvable only if the set P localizes all X. For example, if we place beacons at all candidate locations, all floor plan points are localized. When this condition is meet, we can have several possible configurations that localize the entire floor plan. Our goal is to find a minimum set B such that all points in X are localized by the beacons located at B.

3.4 Coverage-aware beacon placement

Key to our approach is leveraging the beacon coverage. Below, we define the terms associated with coverage. The coverage map is the mapping between beacons and the locations covered by the beacons, or vice versa.

Definition 3.4. Locations in coverage of a beacon $C_B(b), b \in B$: Set of all locations in the floor plan that are covered by the beacon b.

$$C_B(b) \subseteq X \tag{3.7}$$

Definition 3.5. Beacons in coverage of a location $C_X(x), x \in X$: The individual beacon's coverages are combined to get the set of beacons in coverage of a point.

$$C_X(x) \subseteq B \tag{3.8}$$

Definition 3.6. Zone z_i : Given all points in the floor plan X and the set of candidate locations B with the coverage map, the floor plan gets partitioned into disjoint regions such that all points in the same regions are covered by exactly the same set of beacons. We call these disjoint regions "zones."

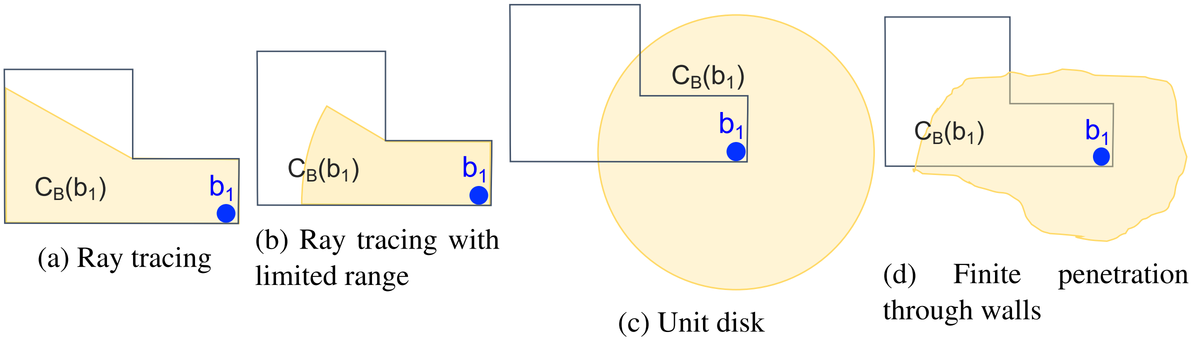
$$Z = z_1 \cup z_2 \cup \cdots \cup z_{N_Z} \tag{3.9}$$

$$\forall z_i, z_j, i \neq j, z_i \cap z_j = \emptyset \tag{3.10}$$

$$z_i = \{x_i \in X | C_X(x_i) = C_X(x_j), x_j \in X\} \tag{3.11}$$

$$C_Z(z_i) = C_X(x_i), x_i \in z_i \tag{3.12}$$

Now that we have defined coverage, next we describe how to generate the coverage models.



(a) Ray tracing

(b) Ray tracing with limited range

(c) Unit disk

(d) Finite penetration through walls

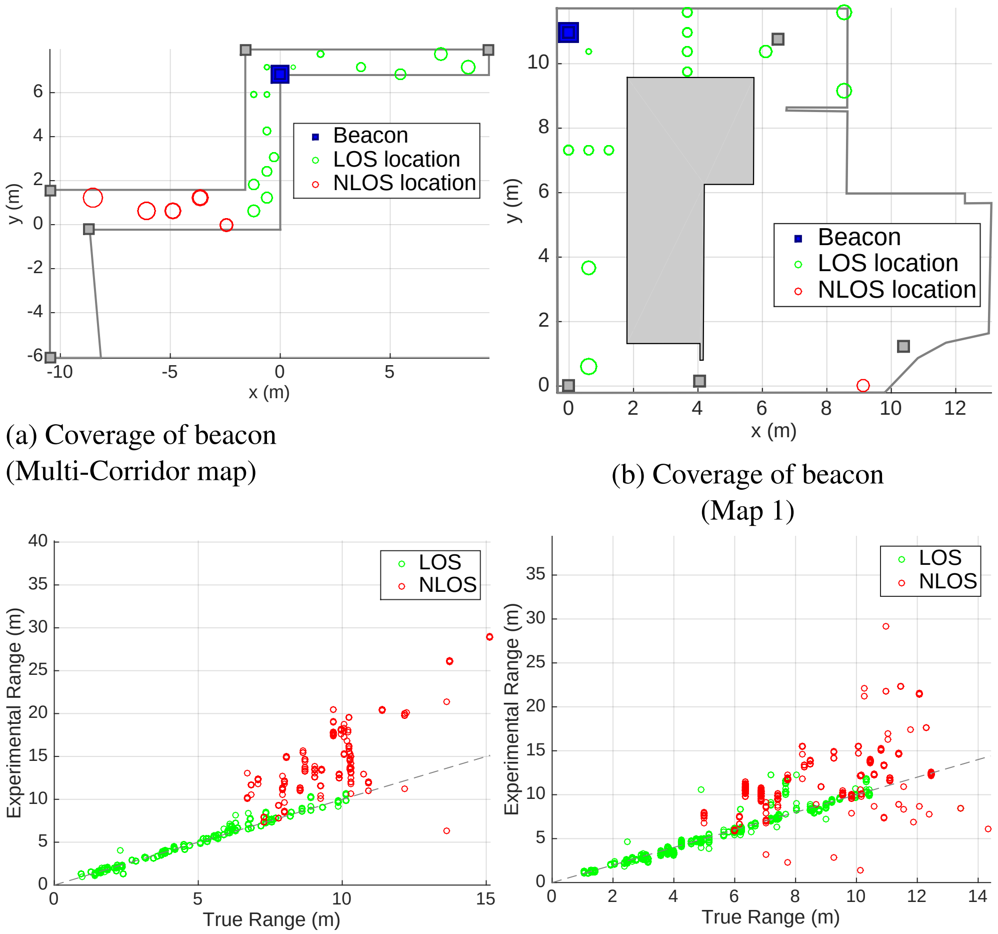
Figure 3.1: Types of beacon coverage models

3.4.1 Models and assumptions

In this section, we elaborate on the models and assumptions of our proposed placement technique that leverages beacon coverage models and knowledge of the floor plan. First, we assume that the floor plan impacts and determines the beacon coverage. We elaborate on this in Section 3.4.1. Among the possible coverage types, we assume the ray-tracing coverage for our analysis. This is elaborated on in Section 3.4.1. We also make assumptions on the LOS noise model while incorporating localization accuracy into the placement algorithm. The experimental validation of the LOS noise model is discussed in Section 3.4.1. Finally, in Section 3.4.1, we discuss the validity of our assumption of a 2D world rather than 3D.

Floor-plan based beacon coverage

Beacon coverage is determined by the interaction of the signals from the beacons with the floor plan. Figure 3.3 shows four types of coverage models. Figure 3.1a shows a ray tracing coverage model common to acoustic/ultrasonic ranging systems. Here, the signal does not penetrate through walls. This model can be extended to RF-ranging systems where low-level signal properties can distinguish between a direct LOS signal and a NLOS signal through a wall. As a result, we can detect and eliminate the NLOS signals and hence reduce the signal propagation to a ray-tracing model. Figure 3.1b shows a ray tracing coverage model with limited beacon range. Typically the ranging technology would be designed to report ranges only if the SNR is above a certain threshold. This results in a limited beacon range in large spaces. Figure 3.1c shows an arbitrary coverage model where the range reduces when the signal penetrates through walls, as is common for several RF-based technologies. However, modeling this type of coverage given a floor plan is hard, as the signal penetration depends on the properties of the barriers. Figure 3.1d shows a unit disk coverage model that is commonly used for analysis, but in practice it is rare to have this type of coverage. Contrary to the other models, this model assumes that the signal penetration is unaltered through walls. In our implementation, we assume the ray tracing model with and without limited range (Figure 3.1a-Figure 3.1b).



(a) Coverage of beacon (Multi-Corridor map)

(b) Coverage of beacon (Map 1)

(c) Range measurement error (Multi-Corridor map)

(d) Range measurement error (Map 1)

Figure 3.2: Experimental characterization of coverage and range error for ultrasonic beacons

Ray tracing model

The ray tracing beacon coverage model assumes that floor plan limits beacon coverage. As a result, out-of-coverage locations do not receive range measurements from beacons, and locations in coverage receive measurements with low error. In reality, while the system is in use, we receive non-line-of-sight (NLOS) signals (due to signals reflecting off walls or penetrating through walls). We cope with NLOS signals while estimating location. In Chapter 4, we discuss a location solver that can localize using range measurements from low-density LOS and high-density NLOS beacons, with knowledge of the floor plan. With a high enough density of LOS, we can cope with NLOS without using the floor plan by checking for consistency among the LOS measurements [72, 137]. When we have access to other sources of information, such as the low-level properties of the physical signal, other schemes can be used for detecting NLOS range measurements, which we elaborate on in Section 4.1.2. Our ray tracing model is motivated by the fact that in practice the LOS signals are predictable, but the NLOS signal paths are not predictable. We recommend designing the beacon placement for LOS coverage and coping with NLOS signals while estimating location.

LOS ranging noise model

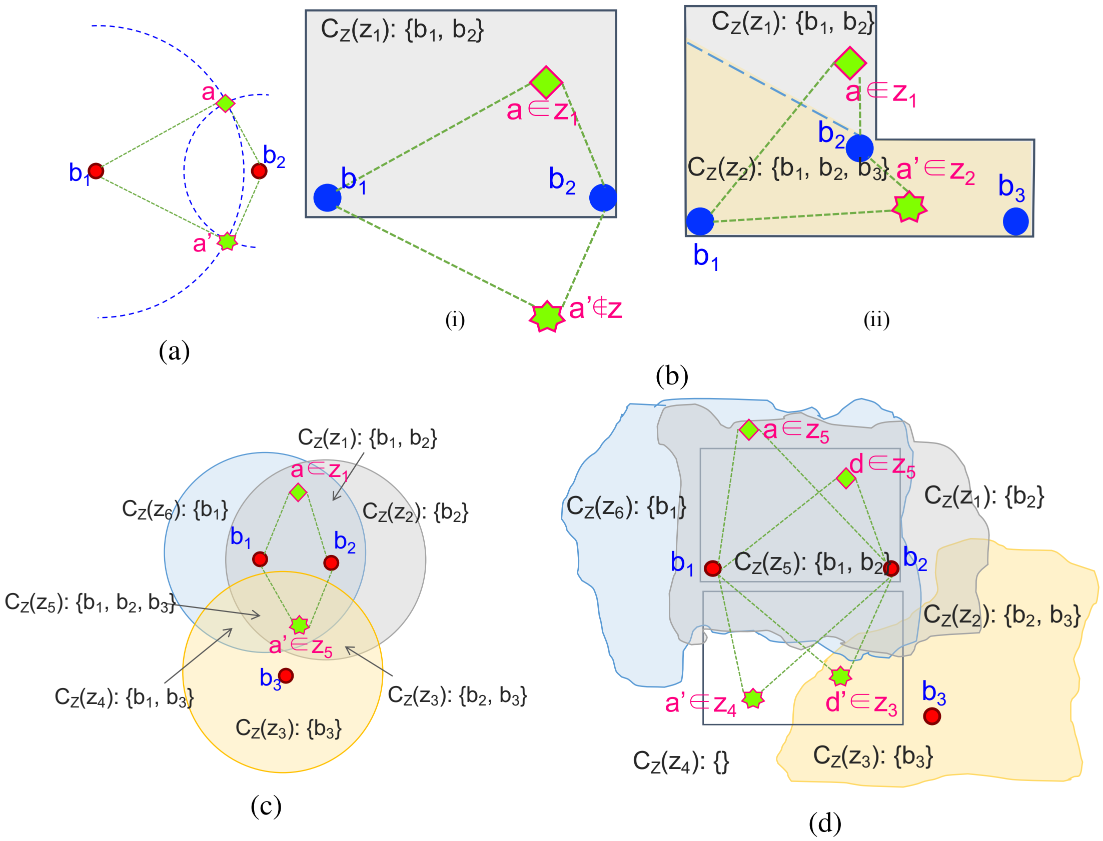
The ray-tracing coverage model determines the regions in and out of the coverage of beacons. In an ideal world, locations in coverage of a beacon would receive range measurements with no error. In reality, range measurements are noisy. For our analysis, we assume that LOS range measurements have zero mean constant variance additive Gaussian noise, and that NLOS range measurements have an additive noise that is positively biased (expected value higher than zero). Figure 3.2c and Figure 3.2d show the distribution of the LOS and NLOS ranges of an ultrasonic ranging system [66] on two floor plans, for 5-6 beacons with at least 500 range measurements taken uniformly across all regions in the floor plan. We observe that when the beacons are in LOS, the range measurements have nearly zero bias and almost constant variance. There is no appreciable change in variance of the range measurements with distance up to $10m$. This is likely due to the SNR being sufficiently high that the distance from the beacon does not affect the ranging accuracy.

Two-dimensional beacon placement

We solve the beacon placement problem in 2D, but in practice, we deploy beacons in 3D space. The natural question that arises is, "Does the 2D beacon placement strategy apply for a 3D deployment?" The proposed approach and concepts extend from 2D to 3D space, allowing the same strategy and tools to be applied for 3D beacon placement. In our experience of deploying these systems in the real world, we see that indoor spaces typically vary in the x-y plane rather than the z plane. If we deploy beacons close to the ceiling and the target device is held by a user nominally around 1m from ground, a 2D model holds as most temporary obstructions in the environment, e.g., chairs and tables, would not impact the direct signal between the beacon and target device. However, the 2D model assumption would not hold for a 3D deployment when beacons are deployed at floor level and blocked by objects, or when beacons are deployed at ceiling level and tall obstructions such as cubicle partitions block the target device held from the beacon. For practical purposes, the beacons can be deployed at ceiling level in most public spaces such as airports, museums and malls, and the 2D coverage will be applicable. The choice of whether to solve for beacon placement in 3D or to solve in 2D and deploy beacons at the ceiling should be made based on whether the end-application requires 2D positioning accuracy or 3D positioning accuracy. For instance, if accuracy in the z-axis is important, the beacons should be well spread out along the z-axis to improve the diversity. However, for applications such as wayfinding indoors, where z accuracy is not critical, a 2D beacon placement approach will suffice.

3.4.2 Unique localization with reduced beacons

Consider the two-beacon scenario in Figure 3.3a, with beacons b_1 and b_2, with the receiver's true location as a. The location a' also receives the same range measurements and thus we cannot disambiguate between a and a'. However, by making use of beacon coverage information, it is possible to disambiguate between the two locations under the condition that the set of beacons providing coverage is different for a and a'. Applying the coverage model of the beacons across



(a)

(i)

(ii)

(b)

(c)

(d)

Figure 3.3: Localizing with two beacons based on beacon coverage. (a) Localization ambiguity with two beacons (b) Ideal ray-tracing: Infinite range, permeability=0 (c) Unit disk coverage: Fixed finite range, permeability=1 (d) Arbitrary coverage: 0<Permeability<1

Class	Color code	Coverage	Uniquely Localizable
1	🟩	$\#C_X(x) \geq 3$	1
2	🟦	$\#C_X(x) = 2$, $C_X(x') = \{\}$	1
3	🟦	$\#C_X(x) = 2$, $C_X(x') \neq C_X(x)$, $C_X(x') \neq \{\}$	1
4	🟪	$\#C_X(x) = 2$, $C_X(x') = C_X(x)$	0
5	🟨	$\#C_X(x) = 1$	0
6	🟥	$\#C_X(x) = 0$	0

Table 3.1: Legend for different coverage classes based on number of beacons in range and unique localization

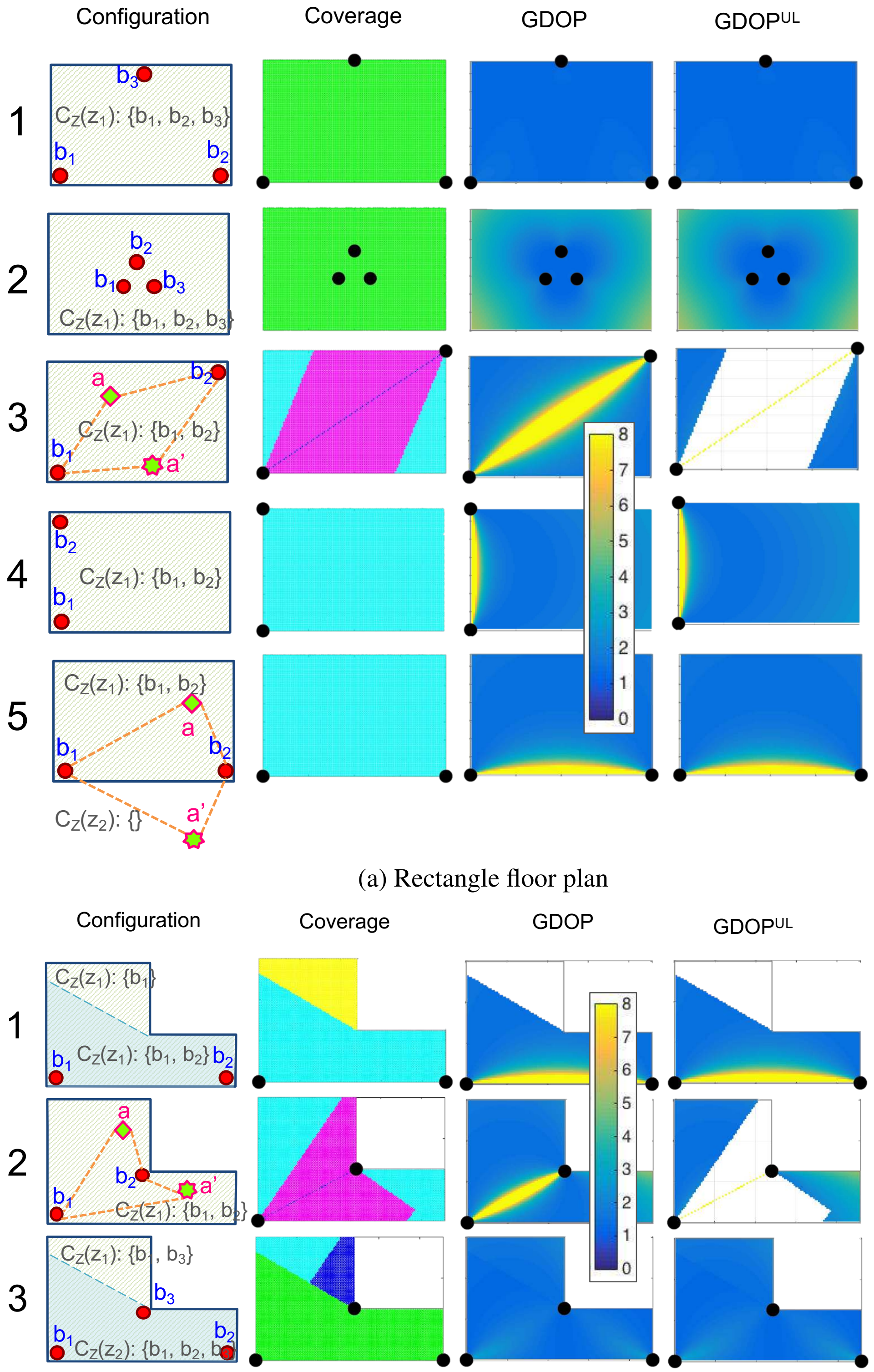
the floor plan partitions the floor plan into distinct zones, each covered by a unique set of beacons, shown in Figure 3.3. Figure 3.3b shows ideal ray-tracing coverage. In Figure 3.3b(i), with the ray-tracing coverage model, we see that the zone inside the floor plan has $C_Z = \{b_1, b_2\}$ and the zone outside the floor plan has $C_Z = \{\}$. If the true location is a and the ranges from b_1 and b_2 are received, we can disambiguate the location a from a', since a' cannot receive measurements from b_1 and b_2. Effectively, a and a' are located in different zones. In Figure 3.3b(ii), both zones z_1 and z_2 are covered by b_1 and b_2, and in addition, z_2 is covered by b_3. Here, though both a and a' receive the same ranges from b_1 and b_2, we can disambiguate them since a' would receive a range measurement from b_3 as well. In both Figure 3.3c (with unit disk coverage) and Figure 3.3d (with arbitrary coverage), we can resolve the location a from a', and d from d', since the two ambiguous locations have a different beacon coverage, i.e., they are located in different zones. *In this manner, if the localization solver incorporates the coverage information of the beacons, we can localize using two beacons, instead of three.* Though a simple concept, this is the key insight that results in an approximately 33% reduction in the number of beacons required across a building floor plan by using our approach for beacon placement.

Definition 3.7. Uniquely Localizable: We define a location to be *Uniquely Localizable* if in the absence of noise, it can be localized without any ambiguity, when range measurements are received from the beacons that provide coverage to the location. We define the function $UL(x, C_X(x)) \in \{0, 1\}$, which has a binary output, as:

$$UL(x, C_X(x)) = \begin{cases} 1, \#C_X(x) \geq 3 \\ 1, \#C_X(x) = 2, C_X(x) \neq C_X(x') \\ 0, \#C_X(x) = 2, C_X(x) = C_X(x') \\ 0, \#C_X(x) \leq 1 \end{cases}$$

where $\#C_X$ denotes the cardinality of the set C_X and x' is the reflection of x about the line joining the two beacons in the set $C_X(x)$. Note that x' is only defined when $\#C_X(x) = 2$. We subsequently use the notation $UL(x)$ instead of $UL(x, C_X(x))$.

Table 3.1 shows the color coding for the beacon coverage that we have used in Figure 3.4. The fourth column shows the value of the binary UL function. Note that the algorithm does not



(a) Rectangle floor plan

(b) L-shaped floor plan

Figure 3.4: Comparison of multiple beacon configurations. Legend for *Coverage* column is in Table 3.1

distinguish between Class 2 and 3 points, but they are shown here for visual purposes. Class 2 points have the ambiguous location outside of the floor plan, and Class 3 points have the ambiguous location inside but covered by different beacons. We see from the second column of Figure 3.4a that *configurations 1, 2, 4* and *5* have all locations Uniquely Localizable either due to coverage by three beacons, or in the case of two beacons, the ambiguous location is outside of the floor plan. In the case of Figure 3.4b, *configuration 3* has all locations Uniquely Localizable. In ideal scenarios, any of the configurations that Uniquely Localize the entire floor plan would provide exact location estimates, and the configurations with fewer beacons then would naturally be desirable. However, in realistic scenarios, these configurations would not all provide the same location accuracy across the floor plan. When range measurements are noisy, the location accuracy is dependent on the error in ranging as well as geometry between beacons and the receiver, as elaborated on in the next section.

3.5 Incorporating localization accuracy with Geometric Dilution of Precision (GDOP)

A useful guideline to quantify the location accuracy is the Cramér-Rao Bound (CRB), the lower bound on the variance in the location that can be achieved by an unbiased location estimator [57]. The results presented in this section are derived from [121], [94] and [93]. For 2D trilateration systems, it has been shown in [121] that with an unbiased estimator, the CRLB variance of the positional error $\sigma^2(x)$ at location x, as defined by $\sigma^2(x) = \sigma_x^2(x) + \sigma_y^2(x)$ is given by:

$$
\begin{aligned}
\sigma^2(x) &= \frac{\sum_{k=1}^{N_c} \sigma_{r,i}^{-2}}{\sum_{k=1}^{N_c-1} \sum_{j=k+1}^{Nc} \sigma_{r,k}^{-2} \sigma_{r,j}^{-2} A_{kj}^2} \\
A_{kj} &= |\sin(\theta_k - \theta_j)|
\end{aligned}
$$

where $\sigma_{r,k}^2$ is the variance in range measurement of beacon k, N_c is the number of beacons in $C_X(x)$ and θ_k is the angle between b_k and x.

Under the assumption that the range measurements are independent and have zero-mean additive Gaussian noise with constant variance σ_r^2, this reduces to:

$$
\begin{aligned}
\sigma(x) &= \sigma_r \times \sqrt{\frac{N_c}{\sum_{k=1}^{N_c-1} \sum_{j=k+1}^{N_c} A_{kj}}} \\
\sigma(x) &= \sigma_r \times GDOP(x)
\end{aligned}
$$

The GDOP [81] is a function of the angles between the target x and beacons $C_X(x)$, and is given by :

$$
GDOP(x) = \sqrt{\frac{N_b}{\sum_{k=1}^{N_b-1} \sum_{j=k+1}^{N_c} A_{kj}}}
$$

The CRB is directly proportional to the GDOP, and as seen in Section 3.2, several authors have used GDOP to quantify the location accuracy. We are using the notation $GDOP(x)$ instead of $GDOP(x, C_X(x))$. The third column ***GDOP*** in Figure 3.4 shows a GDOP map of six representative floor plans. The GDOP is the worst (highest) along the line joining two beacons, and is in general better (lower) when the regions are covered by more beacons. As a numeric example, for a ranging system with standard deviation of the range measurements of $10cm$, if two beacons subtend an angle of $90°$ at a target location, the resulting GDOP is 1.414 and the 2D location estimate would have a standard deviation of $14.14cm$.

3.5.1 A modified GDOP metric for indoors

We now formalize our modified GDOP metric to use Unique Localizability, which is given by:

$$GDOP^{UL}(x) = \begin{cases} GDOP(x), & UL(x) = 1 \\ NaN, & UL(x) = 0 \end{cases}$$

The fourth column of Figure 3.4 shows the $GDOP^{UL}$ metric. For most configurations, it is the same as the GDOP metric. But for *configuration 3* of Figure 3.4a and *configuration 2* of Figure 3.4b, where there exist *Class 4* locations with the two-beacon ambiguity problem, the $GDOP^{UL}$ is not defined where Unique Localization cannot be achieved. These cases are now numerically handled to avoid providing a confidence on the location estimate when ambiguity exists in the solution.

3.5.2 Quality of a beacon configuration (Q)

With our modified GDOP metric, we can now compare beacon configurations like those found in Figure 3.4a and Figure 3.4b. We utilize this in our toolchain which has two modes of operation, where it can either optimize for Unique Localizability or optimize for both Unique Localizability and GDOP.

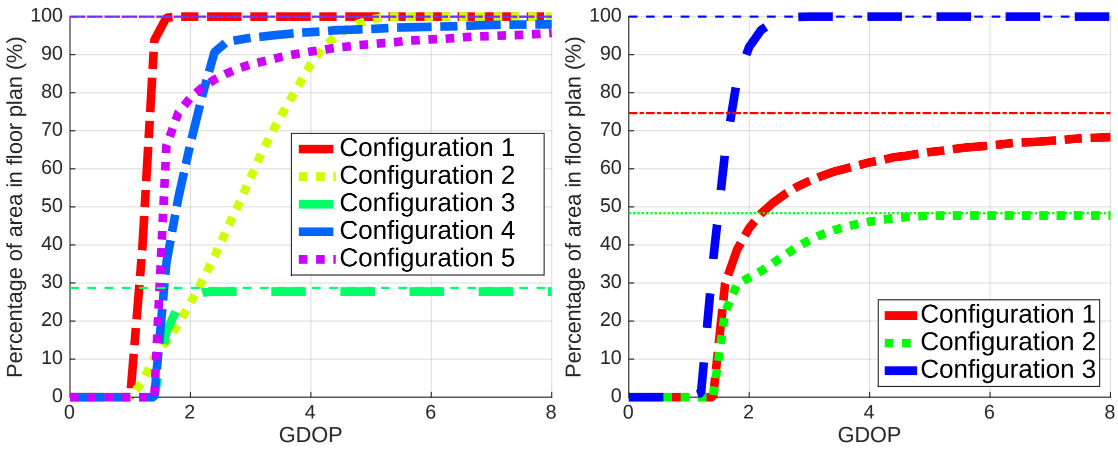
Case 1: UL-based metric Q_{UL}: In this case, we attempt to place beacons such that all regions in the floor plan are Uniquely Localizable without considering the localization accuracy. This could be required in an ideal scenario with no ranging noise where the geometry of the beacons does not affect the localization accuracy, or when the location estimate is averaged over a large number of measurements, resulting in low variance. We define the quality of the beacon configuration B across the floor plan X, as the percentage of area that is Uniquely Localized.

$$Q_{UL}(X, B) = \frac{\sum_{i=1}^{N_x} UL(x_i)}{\#X} \times 100$$

where $UL(x_i)$ is the binary function that indicates if a location x_i is Uniquely Localizable when covered by the beacons $C_X(x_i)$, as defined in Section 3.4, and $\#X$ is the cardinality of set X or number of points in the floor plan.

In Figure 3.4a, Q_{UL} is 100% for *configurations 1, 2, 4* and *5* and 28.7% for *configuration 3*. In Figure 3.4b, *configurations 1, 2* and *3* have Q_{UL} as 74.6%, 48.3% and 100% respectively.

Case 2: UL and GDOP-based metric $Q_{GDOP^{UL}}$: The $GDOP^{UL}$ metric in Section 3.5.1 is

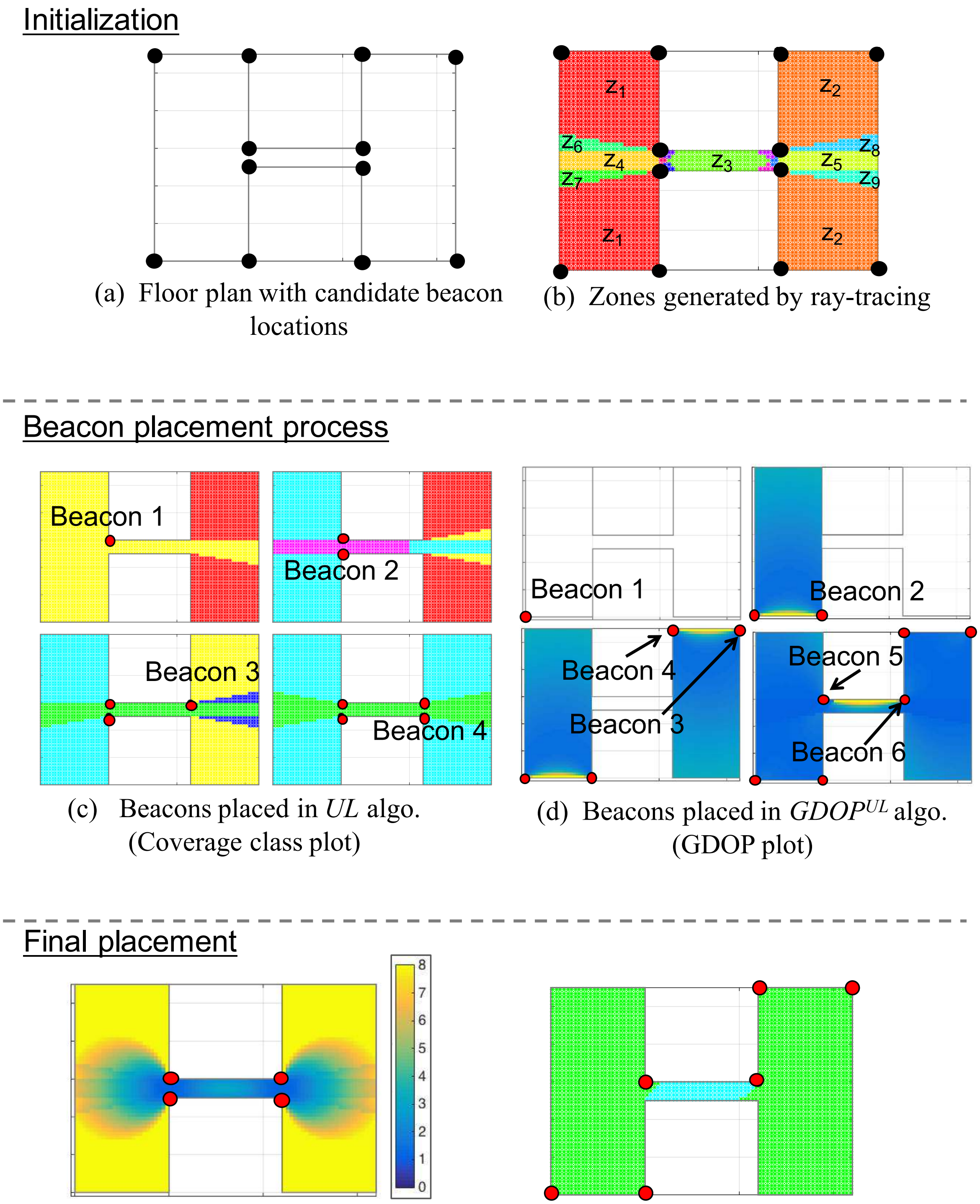


(a) Rectangle floor plan

(b) L-shaped floor plan

Figure 3.5: Comparing configurations using *quality of beacon configuration* $Q_{GDOP^{UL}}$ for the configurations in Figure 3.4

defined for a single location. In order to quantify the quality of the beacon configuration across all locations in the floor plan, we use a heuristic based on the Cumulative Distribution Function (CDF) of the $GDOP^{UL}$ curve across all locations, as shown in Figure 3.5. For instance, to compare *configuration 1* and *configuration 2* of the rectangular room, where both configurations have three beacons, we see from Figure 3.5a that 60% of the floor plan has a GDOP less than 1.7 under *configuration 1*, and a GDOP less than 3.0 under *configuration 2*. Alternately, 100% of the floor plan has a GDOP under 3.0 in *configuration 1* but only 60% of the floor plan has a GDOP less than 3.0 under *configuration 2*. Hence *configuration 1* is better, since it has a lower overall GDOP than *configuration 2*. For *configuration 3*, where a large part of the floor plan is not Uniquely Localizable, we can see from the CDF plot that only around 27% of the floor plan is Uniquely Localized. We can see from these curves that *configuration 1* is better than *configuration 2* which is better than *configuration 3*. However, it is not obvious how *configuration 4* and *configuration 5* compare since the CDF curves intersect. If the goal is to have 60% of the floor plan with lower GDOP, *configuration 5* is better, but if the goal is to have 90% of the floor plan with lower GDOP, then *configuration 4* is better. In our toolchain, the designer can specify the requirement, but by default we define $Q_{GDOP^{UL}}$ as the area under the $GDOP^{UL}$ CDF curve. To compute the area, we need to provide an upper limit on the GDOP. For the plots shown, the upper limit is conservatively chosen to be 8.0, which corresponds to an angle of $1.8°$ between two beacons and a target. This is equivalent to considering the regions with GDOP worse than 8.0 to not be localizable. For the L-shaped room in Figure 3.4b, we see from Figure 3.5b that among the configurations with two beacons, *configuration 1* is better than *configuration 2* even though *configuration 2* has better coverage, and *configuration 3* with three beacons outperforms both these configurations. The same metric can be used even if the target area is a predefined path or a finite set of locations across the floor plan. In these cases, the CDF would be computed only over the desired target locations.



(a) Floor plan with candidate beacon locations

(b) Zones generated by ray-tracing

(c) Beacons placed in *UL* algo. (Coverage class plot)

(d) Beacons placed in $GDOP^{UL}$ algo. (GDOP plot)

(e) *UL* algo. (GDOP plot)

(f) $GDOP^{UL}$ algo. (Coverage class plot)

Figure 3.6: Step-by-step results of the beacon placement process (Legend for coverage plots is in Table 3.1)

3.6 Beacon placement algorithm

In this section, we present a beacon placement algorithm based on the concepts described in Section 3.4 and Section 3.5. The inputs provided are the floor plan, the coverage model of the beacons and the set of candidate beacon locations. The floor plan and obstacles are represented as multiple polygons. The implementation is illustrated by Figure 3.6 for a floor plan that represents two rooms connected by a corridor. Figure 3.6(a) shows the floor plan with candidate beacon locations shown by black circles. We also designed a tool where the user can draw floor plans and mark candidate beacon locations to aid in prototyping.

The algorithm has two modes of operation, based on whether we want to optimize based on Q_{UL} or $Q_{GDOP^{UL}}$. Our tool provides several design options to the user, such as placing beacons until a finite number of beacons are placed or placing beacons until some stop criteria is satisfied. This stop criteria could be in the form of accuracy requirements across the floor plan, for instance $GDOP \leq 4.0$ in 90% of the region. The stop criteria we have used for both modes of the algorithm is achieving Unique Localization across the entire floor plan. We describe the algorithm below.

Step 1: Initialization

- Discretize the floor plan to generate X.
- Apply the beacon coverage model for each candidate location b_j to generate the coverage of the beacon configuration.
- Partition the floor plan X into zones z_i such that all x_i that have the same beacon coverage, $C_X(x_i)$, belong to one zone z_i.
- For each zone, assign:
    - *Localization Status* $= 0$
    - *Size* = number of points in the zone
- Initialize *Selected Beacons* = $\{\}$

Figure 3.6(b) shows the zones generated by ideal ray-tracing coverage with five of the largest zones labeled.

Repeat Steps 2-4 until *Localization Status* of all zones=1. In every iteration, one beacon is placed.

Step 2: Select zone

Among the zones with *Localization Status* = 0, select the zone with largest *Size*

Step 3: Select subset of candidate locations

Among all candidate beacon locations P, select the subset $C_X(x_i)$, where x_i is any point in the zone. Note that all points in the zone are covered by the same candidates.

Step 4: Among the subset of candidates, select the candidate that maximizes the criteria given below:

- The selection criteria depends on whether we are optimizing for UL or $GDOP^{UL}$. Further, since these metrics are not defined for single-beacon cases, we have a different criteria when N_b, the number of beacons already covering the zone, is zero.

(1) UL *and Nb=0*: Select the candidate with maximum coverage.

(2) $GDOP^{UL}$ *and Nb=0*: In order to provide good geometry, select the candidate with the max-

imum average distance from other beacons for this zone, that are present in *Selected Beacons*.
(3) UL *and* $Nb \neq 0$: Select the candidate that maximizes the *quality of beacon configuration* Q_{UL} or the percentage of area Uniquely Localized. To compute this, use the beacons in *Selected Beacons*.
(4) $GDOP^{UL}$ *and* $Nb \neq 0$: Select the candidate that maximizes the *quality of beacon configuration* $Q_{GDOP^{UL}}$ or the area under the CDF of the $GDOP^{UL}$ curve. To compute this, use the beacons in *Selected Beacons*.

- Add selected candidate beacon location to the set *Selected Beacons*.

Figure 3.6(c) shows the placement of beacons in the UL mode with the coverage classes shown, and Figure 3.6(d) shows the placement of beacons in the $GDOP^{UL}$ mode with the $GDOP^{UL}$ values shown. As can be seen, the selection of the first beacon (to localize the zone labeled 1 in Figure 3.6(b)) and subsequent beacons is different for both the algorithms.
Step 5: Re-evaluate zones

- It is possible that a zone is partially localized by the ambiguity being resolved when the new beacon is placed. In that case, split it into two zones before the next step. Assign *Size* and *Localization Status* for the new zones.
- For all zones: If all the points in the zone satisfy the stop criteria (achieving Unique Localizability), assign *Localization Status* of zone=1.

As we can see from Figure 3.6(c), the entire floor plan is localized with only four beacons while optimizing for Unique Localization. However, the beacons are clustered close together and the GDOP of the final beacon placement, shown in Figure 3.6(e), is poor. On the other hand, Figure 3.6(d) shows the placement while optimizing for GDOP as well, and places two additional beacons. The final configuration has a good GDOP and coverage across the floor plan, as seen in Figure 3.6(d) and Figure 3.6(f) respectively. If the design requirement was to only place four beacons, the first four beacons would have been placed, with the corridor area not localizable but with good GDOP in the two rooms.

3.7 Beacon placement prototyping toolchain

We implemented the proposed algorithms in a MATLAB-based toolchain. We considered different approaches to generate floor plans, select initial location candidates and implement beacon placement algorithms.
(1) Floor plan generation: We implemented these types of floor plans:

- User drawn floor plan through our GUI on MATLAB
- Randomly generated simple polygon with user-defined number of vertices
- A pre-defined floor plan. This option is for providing real-world floor plans as inputs.

(2) Candidate beacons: The toolchain allows the user to select parameters of the candidate beacons. The first parameter is the beacon range. The range can be infinite (*Inf*), in which case it is only limited by the floor plan boundaries, or can be finite, specified in meters. The second parameter is the beacon location. The beacon locations can be vertices or interior points. The interior points can further be randomly generated or user-specified through the MATLAB GUI.
(3) Beacon placement algorithms: We implemented these placement schemes in the toolchain:

Floor Plan	**Beacon Placement Method**			
	3-beacon Minimal	**UL Minimal**	**UL Algorithm**	$GDOP^{UL}$ **Algorithm**
Rectangle	3	2	2	2
L-shape	4	3	3	3
Spokes	5	2	2	2
Two-rooms	6	4	4	6
Multi-room	9	6	8	8
Multi-corridor	7	5	5	5
Map 1	11	8	8	9
Map 2	12	8	9	10
Map 3	9	7	7	8
Map 4	21	15	16	16

Table 3.2: Performance of proposed algorithms in terms of number of beacons placed

- User can select beacon locations through GUI
- **UL Algorithm:** Proposed approach for coverage optimization.
- $GDOP^{UL}$ **Algorithm:** Proposed approach for accuracy optimization.
- **UL minimal:** This is the optimal that our algorithm aims to achieve. We obtain this by searching through all possible solutions.
- **3-beacon minimal:** Optimal solution where any points in the domain are covered by at least three beacons. This represents the best-case scenario with typical state-of-the-art placement methodology.

3.8 Evaluation

We evaluated our beacon placement algorithm in simulations on 10 floor plans, which are listed in Table 3.2. The first five are smaller floor plans constructed to represent different geometries that could occur within larger floor plans. The next four are real-world floor plans in buildings on our university campus, and the last is a floor plan drawn using our toolchain. The *Rectangle* and *L-shape* floor plans are shown in Figure 3.4, the *Two-rooms* floor plan is shown in Figure 3.6 and the remaining seven floor plans are shown in Figure 3.7.

The first column of Table 3.2, *3-beacon (Minimal)* denotes the minimal number of beacons if all regions are to be covered by at least three beacons as required by trilateration. The second column *UL (Minimal)* denotes the minimal number of beacons required under the proposed scheme where we localize with two beacons. The results for both of these columns are obtained using brute-force by iterating through every possible beacon configuration. Note that this is not practical for larger floor plans. We see that the proposed scheme results in reducing the number of beacons by 22-60$\%$ (33$\%$ on average) across these floor plans as compared to typical approaches that provide three-beacon coverage in all areas. The most significant improvement is seen for the

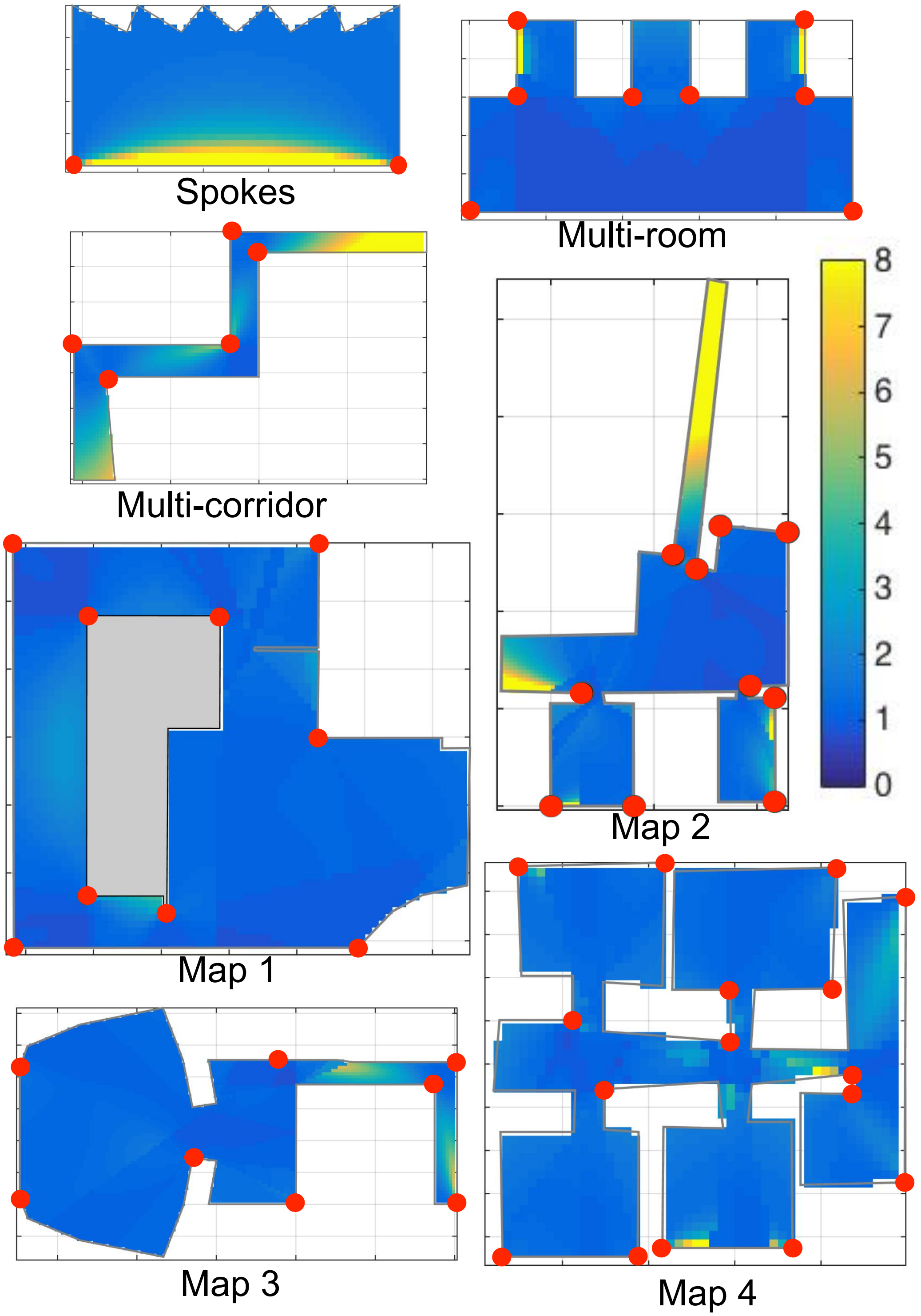
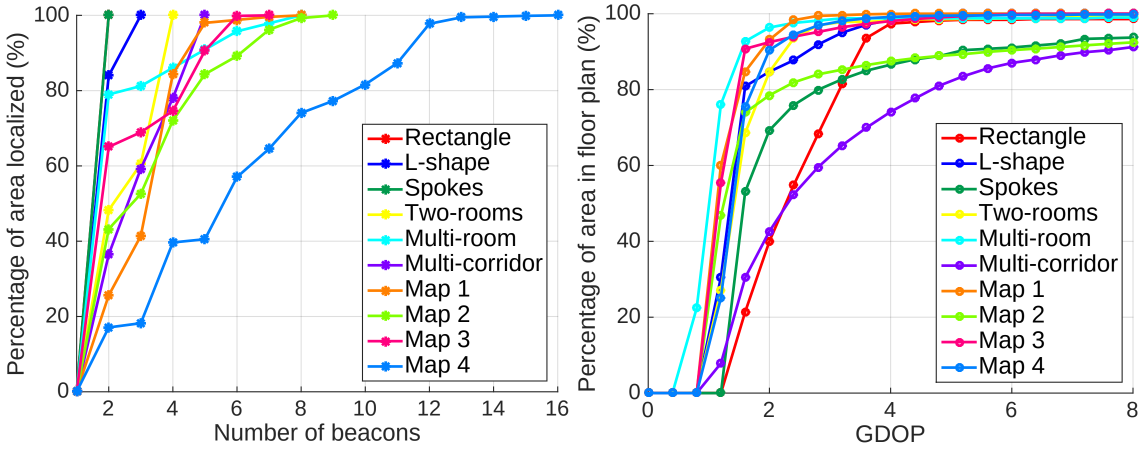


Figure 3.7: Beacon placement result and corresponding GDOP for subset of the floor plans with algorithm optimizing for $GDOP^{UL}$

.



(a) Area localized with addition of every beacon (UL algo.)

(b) GDOP CDF curve for final placement ($GDOP^{UL}$ algo)

Figure 3.8: Validation of algorithm design choices (a) greedy approach and (b) stop criteria

Spokes floor plan, shown in Figure 3.7. This is not typical of indoor spaces, but we selected this to show when the proposed scheme is most effective. The floor plan is generated such that all points are in LOS of the left bottom and right bottom corners. We can extend this floor plan to generate infinite spokes, and the 3-beacon scheme would require infinite beacons, whereas the proposed UL scheme would require only two beacons.

The third column shows the number of beacons placed while optimizing for *UL* and the fourth column shows the number of beacons placed while optimizing for the $GDOP^{UL}$. We see that the UL algorithm places the same number of beacons as the minimal in most cases, with one or two additional beacons in some cases (on average 5% more beacons than minimal). The $GDOP^{UL}$ algorithm usually places the same number of beacons as UL, in worst-case it places two more beacons. In Figure 3.7, we show the resulting GDOP map of the final beacon placement with the algorithm in the $GDOP^{UL}$ mode. We have not shown the minimal beacon placement and result of UL algorithm for these seven floor plans, but the number of beacons for both are shown in Table 3.2.

Figure 3.8a validates the greedy approach we have adopted, where we iteratively select beacons to localize the largest zone. For each of the floor plans, we can see the $\%$ of area Uniquely Localized as an additional beacon is added. For instance, in Map 4, 90% of the area is localized with 11 beacons and an additional five beacons are required for the remaining 10% of the area. For practical purposes, it may be sufficient to have coverage in 90% of the region, since often tracking or filtering would be used in the location solver. Figure 3.8b shows the final CDF of the $GDOP^{UL}$ when the algorithm is in the $GDOP^{UL}$ mode. The final quality of beacon configuration varies across floor plans, since the stop criteria is all regions being Uniquely Localizable. We could also have specified a different criteria based on GDOP, such as 90% of the floor plan having a GDOP less than 4.0. We see that across the floor plans, the *Multi-room* floor plan has the overall highest quality of beacon configuration, but our algorithm placed two beacons more than the minimal number required. The worst quality of beacon configuration is for the multi-corridor floor plan, and we can see from Figure 3.7 that the two end corridors have a high GDOP.

In summary, the proposed scheme based on Unique Localizability places between 22% to 60% (33% on average) fewer beacons than a typical trilateration-based scheme. The proposed algorithm in UL mode usually achieves the minimal placement and on average places 5% more beacons than minimal, but 29% fewer beacons than typical. The proposed algorithm in $GDOP^{UL}$ mode places on an average 14% more beacons than minimal, 23% fewer beacons than typical, and provides a much better quality (GDOP) of beacon placement. The final quality of beacon placement and the number of beacons required varies with the floor plan geometry. The quantitative results apply even for larger floor plans at building-scale since they can be represented as a union of smaller floor plans.

3.9 Summary

This chapter addressed the problem of beacon placement for range-based localization systems. We formulated a new concept of Unique Localization for range-based systems and designed new metrics for quantifying the quality of a beacon placement. We used this metric to design a beacon placement algorithm, which is part of a MATLAB-based open-source toolchain. The proposed placement method reduces the number of beacons on average by 33% while maintaining the same coverage.

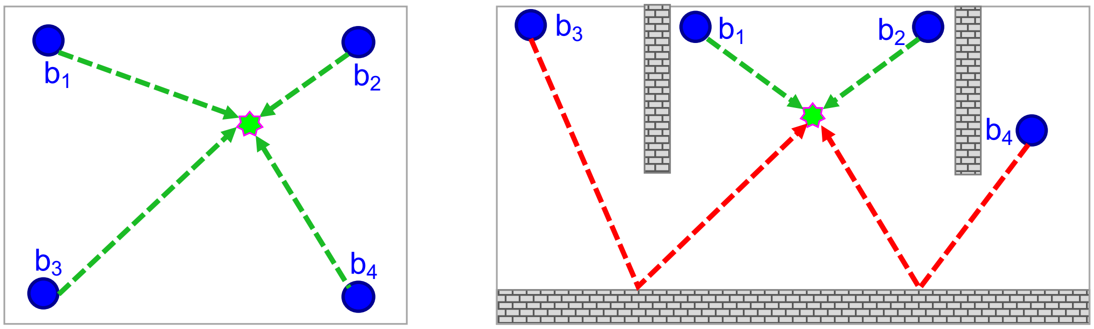
Chapter 4

Location acquisition

In Chapter 3, we discussed strategies for placing beacons to support range-based localization. In this chapter, we present a location solver for range-based systems that is robust to low-beacon density and a high amount of NLOS signals [104].

Though localization with range-based beacons is fairly well understood, prior work in general makes an assumption that there is a high density of beacons and sufficient number of good range measurements. Figure 4.1(a) shows an example deployment with high beacon density and pure LOS that is an ideal scenario often assumed where trilateration is used to estimate location. At lower densities, beacon-based ranging systems struggle to perform accurately due to errors from NLOS signals. Figure 4.1(b) shows a more realistic scenario with NLOS ranges where typical location solvers will fail to produce the correct location estimate. Common solutions to deal with low beacon density and NLOS fuse the information from beacons with inertial sensors or constrain the motion within the floor plan using particle-filters or similar approaches. Unfortunately, these approaches require the user to walk around and explore the space before they can acquire an accurate lock. In many applications such as augmented reality, way-finding and targeted advertising, quick location acquisition is critical for keeping users engaged with the application.

In this chapter, we present an approach that leverages the geometry of the floor plan to both reduce the density of beacons required and to localize in the presence of NLOS. The floor plan information along with the coverage model of the beacons gives us information about which beacons we expect to be in LOS and NLOS in different regions of the building. The proposed location solver (1) considers feasible hypotheses of LOS and NLOS beacons among the received ranges and solves for locations under each hypothesis, (2) checks for consistency between the estimated location and the assumed hypothesis against the predetermined coverage information and (3) selects the most likely hypothesis-location pair. Our approach guarantees that if all expected signals from the LOS beacons are received, then we can maintain the same accuracy even with additional NLOS signals, except in rare deployment geometries. The intuition is that NLOS ranging errors caused by reflected signals are positively biased, and if the location is estimated assuming the hypothesis that an NLOS measurement from a beacon is an LOS measurement, the location will likely be out of the LOS coverage of the beacon, producing an inconsistency with the LOS assumption used to estimate the location. Likewise, a true hypothesis would produce a location estimate that is consistent with the coverage information.



(a) Typically assumed scenario: High beacon density with LOS

(b) Realistic problematic scenario: Low beacon density with NLOS

Figure 4.1: Motivation for a robust location solver

The main improvement of our approach over state-of-the-art systems is that we leverage the coverage information of beacons derived from the floor plan to localize with range-based beacons in the presence of NLOS and low beacon density, thus enabling accurate location acquisition. Our floor-plan aware techniques can be applied to localization systems that have LOS and NLOS error models similar to acoustic systems (zero mean for LOS, positively biased for NLOS), which is the case for multiple emerging TOF RF ranging systems. In support of this approach we design a floor-plan aware solver that is capable of working with low-density beacon deployments and improving robustness to NLOS signals. We also show the design and evaluation of an enhanced ultrasonic speaker array platform that can provide coarse angle information and further prune down the search space and aid the floor-plan aware solver.

In Section 4.1, we cover the background and related work in solvers for range-based localization. This is followed by Section 4.2, where we present the proposed floor-plan aware location solver which is robust to low beacon density and NLOS. We evaluate the approach extensively with real-world experiments and present the results in Section 4.3.

4.1 Background and related work

In Section 4.1.1, we describe existing techniques for trilateration or location acquisition. Next, in Section 4.1.2, we describe techniques to cope with NLOS signals. Since our proposed approach leverages the floor plan information, in Section 4.1.3 we discuss existing work in floor plan integration.

4.1.1 Location acquisition

Trilateration is a well-known process of determining a location by using distance measurements from three (or more) reference points at known locations. [23, 40, 42, 79, 99, 114, 120, 122]. While solving the trilateration problem is non-trivial, due to the non-linear nature of the mathematical problem and practical issues arising from errors in the ranging measurements and geometrical arrangement of the points [41], many algorithms have been developed to determine solutions to this problem and to estimate error [41, 78, 85, 140, 147, 148]. We can generally classify trilateration solvers into closed-form solvers and numerical algorithms. The former [78, 140, 148] have the advantage of a lower computational complexity, and their performance is easier to analyze. However, they do not deal well with errors in the measurements that can cause the solver to fail to find a solution. In the presence of measurement errors, it is usually necessary to perform estimates of the location, based on a numerical algorithm [41, 85, 147]. As a baseline to compare with while evaluating our proposed solver, we implement trilateration using two approaches. In the first approach, which we refer to as grid-search, we search for the location with Minimum Mean Square Error (MMSE) with respect to received range measurements. We discretize occupied regions of the floor plan, first performing a coarse search and then further discretizing the region around the estimated location to perform a finer search. The second approach we use is solving for the MMSE location estimation problem by using gradient descent. However, these techniques assume we have a sufficient number of good LOS measurements and cannot detect and eliminate NLOS measurements. Table 4.1 compares various acoustic/ultrasonic localization systems to give us an idea about the beacon density typically assumed and the mechanisms used to cope with NLOS signals. Most of the systems have been demonstrated to work in single room setups and assume a deployment and scenario as shown in Figure 4.1(a), where all regions are covered by three or more beacons. In practice, covering all regions of a building with three or more beacons can render installation and maintenance prohibitively expensive. A low-density deployment where regions are covered by one or two beacons is desirable, but we can no longer uniquely perform trilateration. In addition, these systems suffer from NLOS measurements in multi-room setups as shown in Figure 4.1(b), where there are two LOS measurements and two inaccurate NLOS measurements. If all beacons are assumed to be in LOS, the estimated location would be incorrect if conventional location solvers are used. In contrast, we solve the problem of estimating location using a single set of range measurements from beacons, which can include NLOS ranges.

4.1.2 Coping with NLOS signals

The last column of Table 4.1 shows various techniques adopted by acoustic ranging systems for combating NLOS. Among the approaches in Table 4.1, [45, 95] use variants of peak detection based on empirical methods that worked well for their system in terms of detecting LOS and NLOS. These approaches use the received signal time series. Another approach to the NLOS problem is to extract statistics from a full Channel Impulse Response (CIR) waveform like Kurtosis, RMS delay spread and mean excess delay [74, 119]. Unfortunately, these statistics are highly dependent on the environment. Filtering of measurements over time can be performed, but significantly increases acquisition times and message passing overhead. In contrast, our solution does not assume that the ranging technology has access to the entire CIR or that several measurements are averaged over time.

Residue-based approaches attempt to detect and mitigate NLOS purely based on the range measurements [17, 26, 72]. They iterate through all possibilities of signals and estimate the locations as a function of the residual error derived from each possibility. Examples of functions include the location with least residue, or average of the locations weighed by the residue. Each of these approaches require three or more LOS beacons or the number of LOS ranges to be

System **year (20xx)**	**Ranging** **type**	**Commodity** **phone?**	**Multi** **room?**	**Number** **beacons** **for 2D loc.**	**NLOS resistance**
Bat[11] ’01 Cricket[97] ’05	TOF	no	yes	3	Limited range, high density outlier in time
BeepBeep[95] ’07	TOF	yes (audible)	no	3	Empirical, system specific
SpiderBat[87] ’11	TOF + AOA	no	no	1	Angle consistency
[65] ’12, [75] ’13	TDOA	yes	no	4	None
Guoguo[74] ’13	TDOA	yes	no	9	Channel stats
ALPS[66] ’15	TDOA first, TOF subsequently	yes	yes	4 for first fix, 3 later	Machine learning BLE, acoustic signal statistics
ASSIST[37] ’15	TDOA	yes	no	4	None
[45] ’15	TOF	no	no	6	Empirical, peak finder system specific
[138] ’16	TDOA	yes	no	4	None
Proposed	TOF	yes	yes	1 for corridors, 2 otherwise	Using floor plan

Table 4.1: Comparison of the conditions under which existing acoustic/ultrasonic localization approaches are evaluated, along with methods to cope with NLOS signals

much higher than NLOS in order for only the true location to produce the least residue. They fail in low beacon density deployments with high NLOS. This motivates the need to develop location solving techniques to cope with localizing in scenarios such as the case in Figure 4.1(b), where all received range measurements are treated with equal confidence. In contrast to these approaches, we cope with NLOS signals by integrating the floor plan geometry to prune out incorrect locations.

4.1.3 Integration of floor plan into location solvers

Several prior works have integrated building floor plans into localization systems. They broadly do so with Bayesian estimation methods by eliminating possible hypotheses of locations over a period of time as the user traverses the indoor space. They exploit the asymmetric nature of indoor spaces and traces walked. For instance, a particle-filter based approach that integrates the information of walls and doors can eliminate unlikely positions of the user [48]. Another approach is to discretize the floor plan, apply a probabilistic model for transitioning between two locations in the floor plan and integrate the motion with a particle filter [49, 141]. These approaches are not suitable for the acquisition problem since they converge on a location estimate over extended periods of time. In contrast, the problem we are solving is estimating location using a single set of range measurements from beacons without having the user walk.

Another class of floor-plan based schemes use complex 3D ray-tracing (accounting for direct, reflected, transmitted, diffracted path, dielectric constant of materials) to model the signal propagation [53, 54, 127]. In contrast, in our work, we use ray-tracing to determine the likelihood of location being in direct LOS or not direct LOS (NLOS) from a beacon. When a NLOS signal is received, the signal can reflect off any of the walls, the floor or ceiling or any other obstruction not modeled by the floor plan. We do not make assumptions on the path of the reflected signal, only when a NLOS path has a positive bias error.

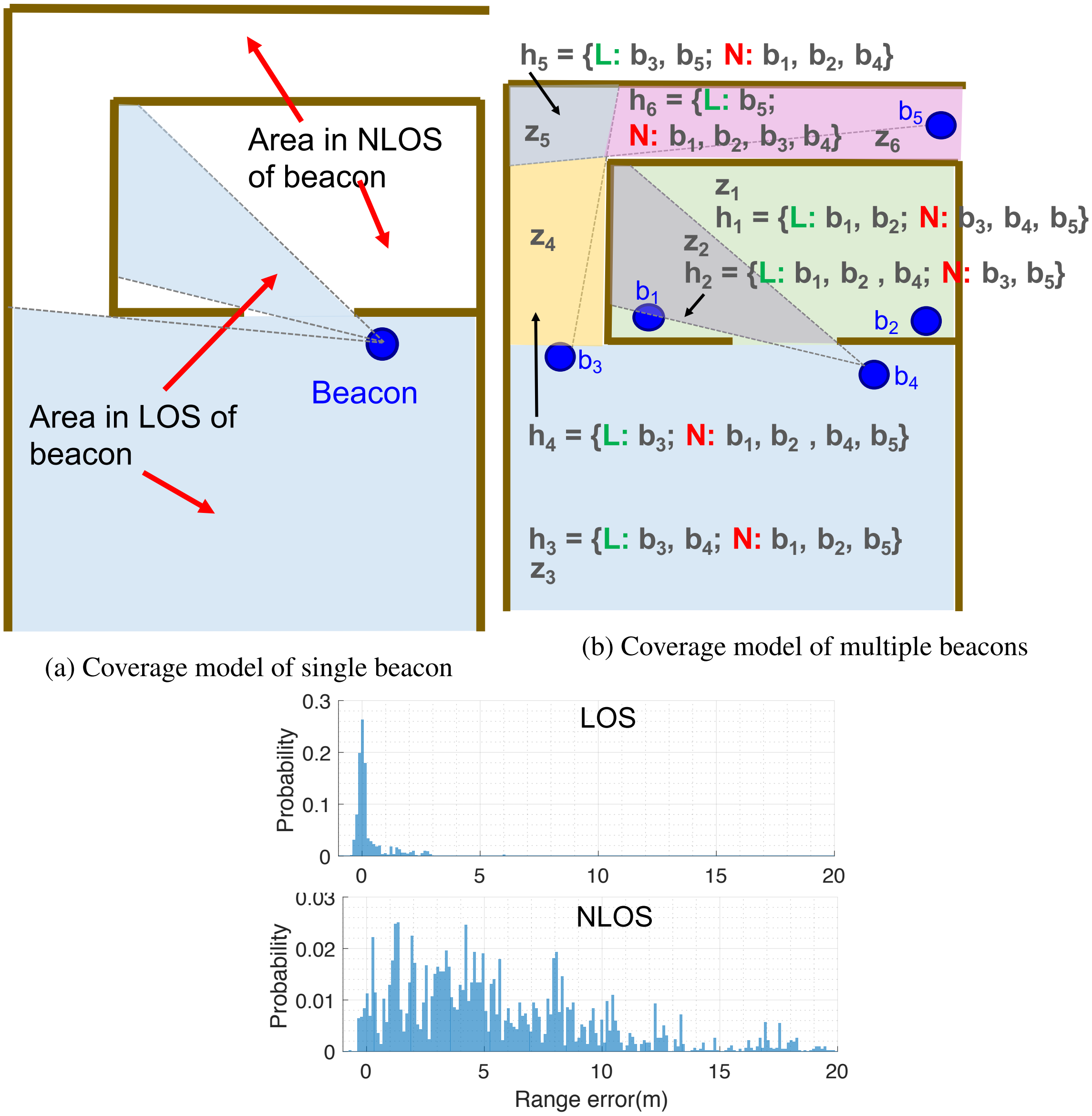
4.2 Coverage-aware location solver

In this section, we describe how the floor plan information can be integrated with the location solver in order to localize in the presence of low beacon density and NLOS signals. The acquisition problem is defined as follows: Given a set of range measurements $[B_r = \{b_i, b_j, ..b_k\}, R = \{r_i, r_j, ..r_k\}]$, where r_i is the range received from beacon b_i, we need to estimate the location of the device receiving ranging signals. For instance, in the two scenarios shown in Figure 4.1, we are given the set of range measurements from the four beacons and must determine the most likely location of the receiver.

4.2.1 Model and Assumptions

Assumptions:

This section builds on the model and assumptions defined in Section 3.4.1 of Chapter 3, with a key difference. In Chapter 3, while designing a beacon placement strategy, we assumed a perfect ray-tracing model where LOS ranges are received for locations in coverage and no ranges


Area in NLOS of beacon
Area in LOS of beacon
Beacon
h$_5$ = {L: b$_3$, b$_5$; N: b$_1$, b$_2$, b$_4$}
z$_5$
h$_6$ = {L: b$_5$; N: b$_1$, b$_2$, b$_3$, b$_4$}
b$_5$
z$_6$
z$_4$
z$_1$
h$_1$ = {L: b$_1$, b$_2$; N: b$_3$, b$_4$, b$_5$}
z$_2$
h$_2$ = {L: b$_1$, b$_2$, b$_4$; N: b$_3$, b$_5$}
b$_1$
b$_2$
b$_3$
b$_4$
h$_4$ = {L: b$_3$; N: b$_1$, b$_2$, b$_4$, b$_5$}
h$_3$ = {L: b$_3$, b$_4$; N: b$_1$, b$_2$, b$_5$}
z$_3$
LOS
Probability
0.3
0.2
0.1
0
0
5
10
15
20
NLOS
Probability
0.03
0.02
0.01
0
0
5
10
15
20
Range error(m)

(a) Coverage model of single beacon

(b) Coverage model of multiple beacons

(c) Experimental accuracy of model

Figure 4.2: Ray-tracing coverage model applied across a deployment

are received at out-of-coverage locations. In this chapter, we relax the assumption and account for NLOS ranges received at out-of-coverage locations. Further, we account for LOS beacons being blocked and as a result our solver copes with missing range measurements from beacons that are expected to be in LOS. Figure 4.2a illustrates the simple ray-tracing coverage model we assume. The beacons are labeled as b_i, and the walls are represented by solid lines. The coverage information is generated automatically by a ray-tracing algorithm operating on floor plans represented in the form of polygons with holes. The algorithm assigns a point to be in LOS with a beacon if the line joining the beacon and the point does not intersect the floor plan polygon. Figure 4.2c validates the ray-tracing model experimentally using data from real-world environments. It shows the distributions of the ranging error between beacons and test points where each range measurement is classified as LOS or NLOS based on the assumed coverage model. In general, as expected, the LOS errors are about zero and the NLOS errors are positively biased and environment-dependent. The model is not perfect and we see some inaccuracy. First, we see that in the LOS case, there is a slight positive bias in the error due to not capturing all obstructions in the environment, such as cubicle partitions, that are not in the floor plan. Second, in the NLOS case, we see that there are locations with close to zero error since we have not accounted for diffraction around doorways and corners. These differences in the real world and our assumptions reflect in our experimental results and we elaborate on this in Section 4.3.5. In future work, we would like to take into account a more realistic modeling of the environment, but for now this model is simple and closely resembles the true coverage of the beacons.

In summary, we assume the following model:

- A NLOS measurement has a positive error. This is because NLOS signals are caused by reflections which always take a longer path than the true path, as experimentally validated in Figure 4.2(b).
- A LOS measurement has zero-mean error.
- The beacon coverage model is deterministic. A location is either in LOS or NLOS of a beacon.

We extend this ray-tracing model to a multi-beacon deployment by combining the coverage information of the individual beacons. As a result, we get the mapping between all the regions of the floor plan and the beacons in LOS of these regions. Figure 4.2b illustrates the resulting coverage. We see that the floor plan gets partitioned into six disjointed zones based on which beacons are in coverage. For example, the blue color shaded region marked with z_3 with $h_3 = \{L : b_3, b_4; N : b_1, b_2, b_5\}$ is the region where b_3 and b_4 are in LOS and b_1, b_2 and b_5 are in NLOS. This partitioning of the floor plan into *zones* was introduced in in Equation 3.9-Equation 3.12. However, earlier, while defining zones, we defined a zone only in terms of the LOS beacons as we used only the LOS beacons for beacon placement. In this chapter, our modified definition of a zone also accounts for the NLOS beacons, as we use both the LOS and NLOS beacons for location acquisition. Here, we introduce the concept of a *hypothesis*. A hypothesis is a partitioning of the beacon set into LOS and NLOS beacons. Though the number of hypotheses possible for a deployment with N_b beacons is 2^{N_b}, only the hypotheses that correspond to a zone are valid for a given deployment. A hypothesis h_i corresponds to the beacons in LOS of z_i being in LOS, and the beacons in NLOS of z_i being in NLOS. For instance, in Figure 4.2b, we can see that the hypothesis $h_i = \{L : b_3, b_1; N : b_2, b_3, b_4\}$ does not correspond to any zone and hence does not correspond to any valid region of the floor plan.

4.2.2 Intuition for proposed approach

We illustrate our approach with two example scenarios in Figure 4.3, where we are required to localize in low beacon density in the presence of NLOS. This floor plan has a combination of hallways covered by single beacons and open areas covered by two beacons. In the first scenario in Figure 4.3a, two LOS and two NLOS ranges are received. In the second scenario in Figure 4.3b, one LOS and one NLOS range are received. The true location x_{true} and the minimum means square error location assuming all received ranges are LOS signals, x_{minerr}, are marked on the figures. We can clearly see that utilizing all the ranges as LOS results in incorrect location estimates in both scenarios.

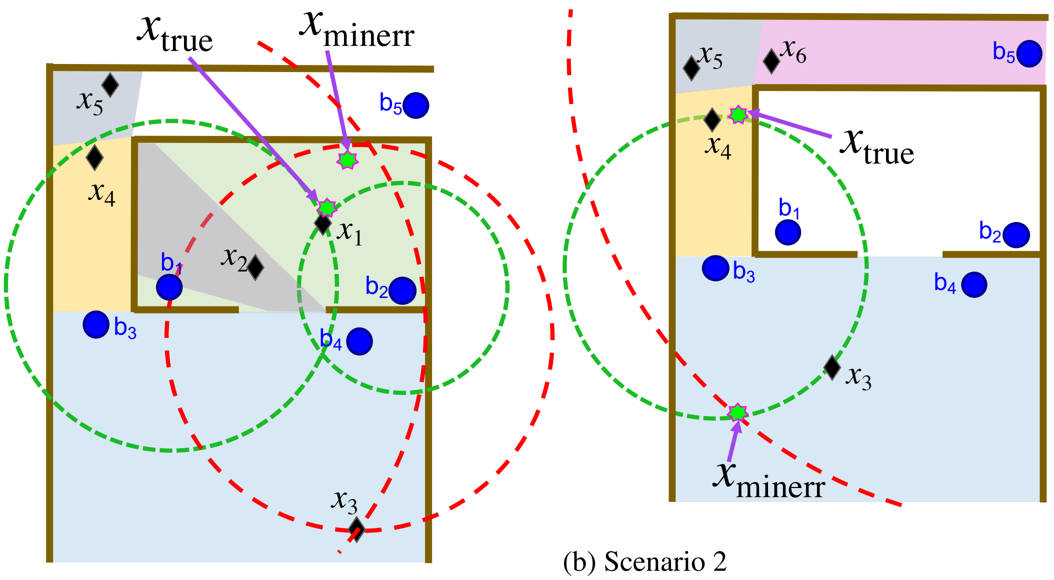
The intuition behind leveraging the floor plan information is that we can check for consistency between the received measurements and the beacon coverage information. In the scenario in Figure 4.3a, the resulting location estimate assuming all the beacons are in LOS is $x_{minerror}$. However, x_{minerr} is in NLOS of b_3 and b_4.This produces an inconsistency with the hypothesis that yielded this location estimate. Another hypothesis we can consider is that b_1 and b_3 are in LOS and b_2 and b_4 are in NLOS. However, according to the coverage information, there is no region of the floor plan that is in LOS of both b_1 and b_3, and hence we can dismiss this hypothesis. Next, say there exists some hypothesis that results in the location being estimated at the point marked as x_4. We see that the distance between x_4 and b_2 is much higher than the range received from b_2, and b_2 is in NLOS of x_4. This implies that the NLOS range measurement has a negative error. This is highly unlikely, given that NLOS measurements have a positively biased error. Hence the hypothesis that estimating x_4 as the true location generates an inconsistency. In the remainder of this section, we show a systematic approach for integrating the coverage information by checking and validating multiple hypotheses to yield a location estimate.

4.2.3 Localization algorithm preliminaries

In this section, we define the notation, state our assumptions and then discuss the conditions that the algorithm uses for checking for coverage consistency.

Notation:

- B: Set of all beacons in the floor plan
- B_r: Set of beacons from which measurements are received
- $\widetilde{B_r}$: Set of beacons from which measurements are not received $B_r \cup \tilde{B}_r = B$
- $B^L(x)$: Set of beacons in LOS of location x
- $B^N(x)$: Set of beacons in NLOS of location x
- h_i: Hypothesis i. A hypothesis is a partition of the beacons into a set of LOS and NLOS beacons.
  $h_i : B = h_i(B^L) \cup h_i(B^N)$
- $h_i(B^L)$: Set of LOS beacons in hypothesis h_i
- $h_i(B^N)$: Set of NLOS beacons in hypothesis h_i



(a) Scenario 1
(LOS ranges: b_1, b_2; NLOS ranges: b_3, b_4)

(b) Scenario 2
(LOS ranges: b_3; NLOS ranges: b_5)

Figure 4.3: Examples to illustrate coverage-aware solver

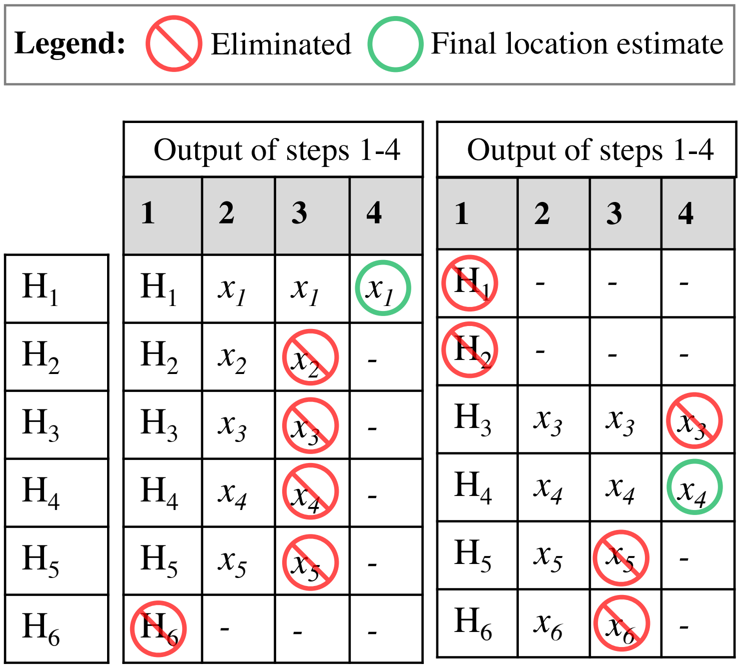
- z_i: Set of all locations in a floor plan that have the exact same set of LOS and NLOS beacons as the partition under hypothesis h_i
- $x_{minerror}$: The minimum mean square error estimated location assuming all beacons are in LOS
- x_{true}: The true location of the receiver

Conditions for satisfying consistency:

If x_i is a location estimated under a hypothesis h_i, in order for this hypothesis-location pair to produce consistency between the beacon coverage model and the received range measurements, the following conditions should be satisfied.

- C_1: Consistency with beacon coverage
  $h_i(B^L) \cap B_r \subseteq B^L(x_i)$
  $h_i(B^N) \cap B_r \subseteq B^N(x_i)$
- C_2: Consistency with NLOS error model
  $\forall k$: $b_k \in h_i(B^N) \cap B_r$, $[r_k - \parallel b_k - x_i \parallel > 0]$

The first two conditions under C_1 check that the LOS/NLOS beacons as assumed by the hypothesis are in LOS/NLOS of location x_i. Since we only receive measurements from the set B_r, we perform an intersection with this set. There can be additional beacons in LOS or NLOS of x_i from which no range measurements are received, hence we use subset and not strict equality in these two conditions. The second condition under C_2 checks that the NLOS errors are positively biased, as per our assumption. We check this condition for all the beacons from which a range is received that are also in NLOS according to the hypothesis h_i.



Legend: 🚫 Eliminated ○ Final location estimate

	Output of steps 1-4				Output of steps 1-4			
	1	**2**	**3**	**4**	**1**	**2**	**3**	**4**
H_1	H_1	x_1	x_1	○ x_1	🚫 H_1	-	-	-
H_2	H_2	x_2	🚫 x_2	-	🚫 H_2	-	-	-
H_3	H_3	x_3	🚫 x_3	-	H_3	x_3	x_3	🚫 x_3
H_4	H_4	x_4	🚫 x_4	-	H_4	x_4	x_4	○ x_4
H_5	H_5	x_5	🚫 x_5	-	H_5	x_5	🚫 x_5	-
H_6	🚫 H_6	-	-	-	H_6	x_6	🚫 x_6	-

(a) Scenario 1 (b) Scenario 2

Figure 4.4: Step-by-step results of proposed *CA-all* solver for the two scenarios in Figure 4.3. The hypotheses are enumerated in Figure 4.2b

4.2.4 Localization algorithm

We describe two location solver algorithms for implementing the approach. The first coverage-aware location solving algorithm evaluates all feasible hypotheses (*CA-all*) and then selects the most likely location-hypothesis pair among the ones that are consistent with the beacon coverage model. While evaluating all hypotheses, in general the hypotheses that assume the short-range beacons to be in NLOS get eliminated. This motivates our second algorithm that ranks all the hypotheses such that a hypothesis with a shorter range LOS beacon gets a higher rank. It then iteratively checks the ranked hypotheses in sequence until one of them satisfies a likelihood criteria. We refer to this as the *CA-short* algorithm, since it prioritizes *shorter* ranges over longer ranges and evaluates fewer hypotheses, and as a result, is *shorter* than the *CA-all* algorithm.

Algorithm 1: ***CA-all*****:**

- **Step 1**: Enumerate all feasible hypotheses.
  The first step is to find all feasible hypotheses. Any hypothesis that includes at least one of the received range measurements as an LOS beacon is a feasible hypothesis. For Scenario 1, since measurements are received from four beacons, b_1-b_4, among the six possible hypotheses, all except h_6 are feasible. For Scenario 2, any hypothesis that includes b_3 or b_5 as an LOS beacon is feasible. The hypotheses h_3, h_4, h_5 and h_6 are selected for Step 2.
- **Step 2**: For each hypothesis in Step 1, find the most likely location that is consistent with the coverage model.
  We then estimate the most likely location for each hypothesis. The location has to satisfy

consistency with the coverage model (consistency check C_1). For each hypothesis, the measurements from the beacons in LOS as per this hypothesis are used to estimate the location with minimum error. While solving under hypothesis h_i, we only need to evaluate the locations that belong to the set z_i. For example, in Step 2 of Scenario 1, while solving for hypothesis h_3, which has only b_3 and b_4 in LOS, any location outside the blue region marked by h_3 would not satisfy either C_1 or C_2. In other words, each location in the floor plan is evaluated only once, i.e., if and when the hypothesis that it satisfies is being evaluated. For hypothesis h_i, the minimum error location estimated is

$$x_i = \underset{x}{\operatorname{argmin}} \left(\frac{1}{N} \sum_k (r_k - \parallel b_k - x \parallel) \right)$$
$$\forall k : b_k \in h_i(B^L) \cap B_r, x \in z_i$$

The black diamond-shaped markers in Figure 4.3 show the most likely estimate for each hypothesis.

- **Step 3**: For each location estimated in Step 2, check if it is consistent with the NLOS error model. Then, we check for consistency with the NLOS error model (consistency check C_2). In Scenario 1, consider the location x_4 under hypothesis h_4. It violates the consistency check with respect to beacon b_2 since $b_2 \in h_4(B^L) \cap B_r$ and the received range measurement $R_2 =\parallel b_2 - x_{true} \parallel$ is less than the distance between x_4 and b_2 ie. $\parallel b_2 - x_4 \parallel$. In the same manner, x_2, x_3 and x_5 are also eliminated. In Scenario 2, at Step 3, hypothesis h_6 with location estimate x_6 and hypothesis h_5 with location estimate x_5 are eliminated.
- **Step 4**: If more than one hypothesis is consistent after Step 3, select the most likely hypothesis based on a likelihood that is a function of the residue LOS error and number of LOS beacons from which no measurements were received.
  Practically, the probability of missing a measurement from an LOS beacon is very low. It would occur if the beacon was temporarily blocked, or if the range packet from the beacon was dropped. Though it depends on the system, we can safely assume that the probability of missing a range from an LOS beacon is lower than the probability of receiving a range from an LOS beacon. Though we cannot estimate this probability since it depends on the environment, we introduce a metric $p(missing_{LOS})$ and assign it a low value to penalize a location with a higher number of missing LOS measurements. We use the number of missing LOS measurements $\#missing_{LOS}$, which is found by the number of beacons in the set $h_i(B^L) \cap \widetilde{B^r}$ for hypothesis h_i. We empirically assign $p(missing_{LOS})$ a value of 0.1 in our implementation. This implies that a location estimate with two missing LOS beacons is assigned a likelihood 0.01 times that of another location with no missing LOS beacons. The second empirical metric we introduce is the residue LOS error. The residue error for location x_i, which is estimated under hypothesis h_i, is:

$$residue_{LOS}(x_i) = \frac{1}{N} \sum_k (r_k - \parallel b_k - x_i \parallel)$$
$$\forall k : b_k \in h_i(B^L) \cap B_r$$

We assign the likelihood of a location x as being proportional to $e^{-residue_{LOS}^2(x)}$, which is equivalent to assuming that the true residue is drawn from a zero-mean Gaussian process. This is the empirical metric we adopt to weigh the LOS residual error and the number of missing LOS beacons. Ideally, the number of beacons used for estimation, the geometry of beacons, the likelihood of a beacon being blocked based on distance from beacon and the environmental factors should be taken into account for accurate modeling.

At the end of Step 3, if we have more than one consistent hypothesis, we assign a likelihood for the locations and select the hypothesis-location pair with highest likelihood:

$$L(x) = \exp^{-residue_{LOS}{}^2(x)} \times \#missing_{LOS}(x)^{p(missing_{LOS})}$$

In Scenario 2, by the end of Step 3, we have two consistent hypotheses x_3 and x_4. Both have zero residue error. Among them, x_3 is less likely since it has a missing LOS range measurement from b_4, and hence we select x_4.

Next, we elaborate on the implementation of the algorithm with the examples in Figure 4.3a and Figure 4.3b. The tables in Figure 4.4a and Figure 4.4b show the results of the algorithm after each of the four steps.

Algorithm 2: ***CA-short*:**

The second algorithm we propose is a variant that is motivated by the observation that several hypotheses get eliminated during Step 3. In Figure 4.3a, we see that beacon b_2 has the shortest range and the four hypotheses that got eliminated in Step 3 were due to producing an inconsistency with this range measurement. This motivates us to sort the hypotheses based on the range measurements, since the hypotheses with the *short*est range beacon in LOS are more likely than the hypotheses with the longest range beacon to be in LOS. In terms of computational complexity, we have the additional step of sorting the hypotheses, but we end up evaluating fewer hypotheses for Step 2 and this algorithm is *short*er than *CA-all*.

- **Step 1a**: Enumerate all feasible hypotheses.
- **Step 1b**: Rank them based on received ranges. We sort the hypotheses such that the hypothesis with the shortest range beacon in LOS has the highest rank. Among them, the hypothesis with second shortest range beacon in LOS has the highest rank, and this repeats. However, we do end up with multiple hypotheses with the same rank. Among them, we give higher rank to the hypothesis with the least number of LOS beacons not in the received range set. The intuition for this is, the locations with a higher number of missing LOS beacons get penalized while estimating the likelihood. Hence, we use this criterion while sorting.
- **Step 2**: For highest rank hypothesis, solve for location.
- **Step 3**: For the location estimated in Step 2, check if it is consistent with the NLOS error model and satisfies the likelihood criteria. If not, then go to Step 2 and evaluate the hypothesis next in rank, and iterate until the stop criteria satisfied. In our implementation, we assigned a value of $L(x) = 0.91$ as a stop condition, since a location with $30cm$ residue and no missing LOS beacons gives a likelihood of 0.91 based on our likelihood function $L(x)$.

According to h_T, $b_k \in$	**According to h_F, $b_k \in$**	**Constraint**
$h_T(B^L)$	$h_F(B^L)$	$\parallel b_k - x_T \parallel = \parallel b_k - x_F \parallel$
$h_T(B^L)$	$h_F(B^N)$	$\parallel b_k - x_T \parallel < \parallel b_k - x_F \parallel$
$h_T(B^N)$	$h_F(B^L)$	$\parallel b_k - x_T \parallel > \parallel b_k - x_F \parallel$
$h_T(B^N)$	$h_F(B^N)$	$\parallel b_k - x_F \parallel > r_k, \parallel b_k - x_T \parallel > r_k$

Table 4.2: Conditions for a false location to be consistent. Both $\{h_T, x_T\}$ and $\{h_F, x_F\}$ should be consistent for the beacons in range

4.2.5 On the robustness of the algorithm

Next, we analyze the robustness of the solver to NLOS signals, under the condition that all expected LOS ranges are received.

We denote the true hypothesis, true location pair as $\{h_T, x_T\}$. To simplify the illustration, we assume zero LOS error. In the presence of LOS error, the same analysis holds with some probability across a set of locations. The true location x_T would be consistent with the beacon coverage corresponding to the true hypothesis h_T by definition. This implies:

$$\forall k : b_k \in h_T(B^L) \cap B_r, [r_k - \parallel b_k - x_T \parallel = 0] \tag{4.1}$$

$$\forall k : b_k \in h_T(B^N) \cap B_r, [r_k - \parallel b_k - x_T \parallel > 0] \tag{4.2}$$

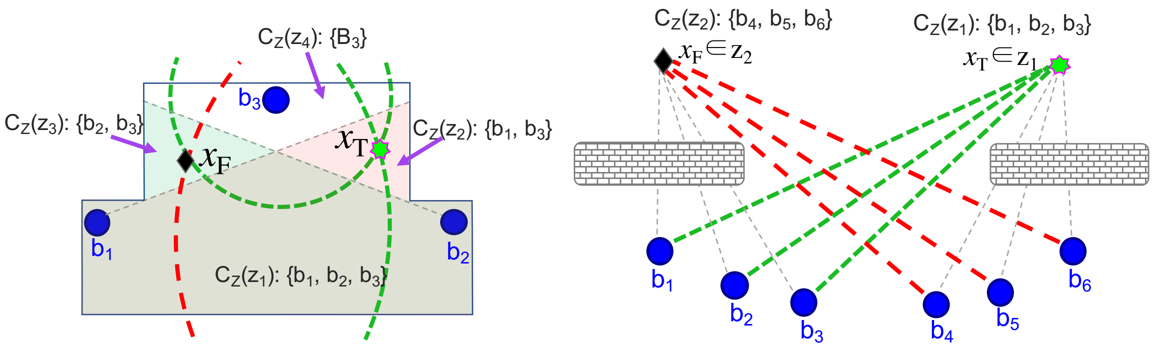
In order for a false hypothesis and false location pair such as $\{h_F, x_F\}$ to be consistent and have the quality to be a feasible solution, the location x_F should produce a consistency between the received measures and the beacon coverage. This implies:

$$\forall k : b_k \in h_F(B^L) \cap B_r, [r_k - \parallel b_k - x_F \parallel = 0] \tag{4.3}$$

$$\forall k : b_k \in h_F(B^N) \cap B_r, [r_k - \parallel b_k - x_F \parallel > 0] \tag{4.4}$$

We want to analyze the conditions under which $\{h_F, x_F\}$ is consistent, given that $\{h_T, x_T\}$ is consistent. We receive measurements only from a subset of beacons B_r. For our analysis, we consider only the beacons from which range measurements are received, as the beacons from which no range measurements are received do not provide us information for this analysis. Among the beacons in range, each beacon is either in LOS or NLOS according to hypotheses h_T and h_F, i.e., under a hypothesis h_i, each either belongs to $h_i(B^L)$ or $h_i(B^N)$. Based on which set it belongs to, a constraint is imposed on the location, given by Equation 4.3-Equation 4.4. For both $\{h_T, x_T\}$ and $\{h_F, x_F\}$ to be consistent, Table 4.2 lists the conditions to be satisfied in the Constraint column $\forall k : b_k \in B_r$. We illustrate the geometrical implications of this with Figure 4.5. In Figure 4.5a, the true location of the device is x_T. There exists another location x_F such that b_3, which is in LOS of both locations, is equidistant from there. Further, though b_1 provides coverage to x_T, x_T is further away from b_1 than a location x_F which is in NLOS but is physically closer to b_1. If b_2 produces a NLOS at x_T, another feasible solution is that the receiver is at x_F and in LOS of b_2 and NLOS of b_1.

If two more beacons are in LOS of a location X_T, then in practice, the constraints on the physical geometry imposed by $b_k \in H_T(B^L)$ make it almost impossible for a false consistent



(a) Example 1

(b) Example 2

Figure 4.5: Illustration of rare scenarios where the solver fails

hypothesis to occur. Figure 4.5a shows an example where this could be possible. Here, x_T is the true location that has received LOS ranges r_1 and r_3 from b_1 and b_3 and an NLOS range r_2 from b_2. Given this, the following conditions are satisfied:

$$r_1 - \parallel b_1 - x_T \parallel = 0 \tag{4.5}$$
$$r_3 - \parallel b_3 - x_T \parallel = 0 \tag{4.6}$$
$$r_2 - \parallel b_2 - x_T \parallel > 0 \tag{4.7}$$

However, there is another location x_F which satisfies the following criteria:

$$\parallel b_1 - x_F \parallel = \parallel b_2 - x_T \parallel \tag{4.8}$$
$$\parallel b_2 - x_F \parallel = \parallel b_1 - x_T \parallel \tag{4.9}$$
$$\parallel b_3 - x_F \parallel = \parallel b_3 - x_T \parallel \tag{4.10}$$
$$\tag{4.11}$$

Given this, x_F is consistent with beacon coverage models, as well as the LOS and NLOS noise models under the assumption that b_2 and b_3 are in LOS and b_1 is in NLOS. Another example is shown in Figure 4.5b, where the true location is x_T with b_1, b_2 and b_3 in LOS and b_4, b_5 and b_6 in NLOS. As is shown, the beacons in LOS of x_T are further away from x_T than the beacons in NLOS. Under the rare geometry condition that the range received by x_T from b_1-b_3 is exactly the same as the distances from x_F to b_4-b_6 respectively and vice versa, x_F also qualifies as a consistent solution. However, these rare failure conditions can be identified during deployment, and we can position the beacons to avoid these cases. A general guideline for placing beacons is to avoid placing them such that they provide LOS coverage to areas much further away from them, while there are large areas close-by in NLOS.

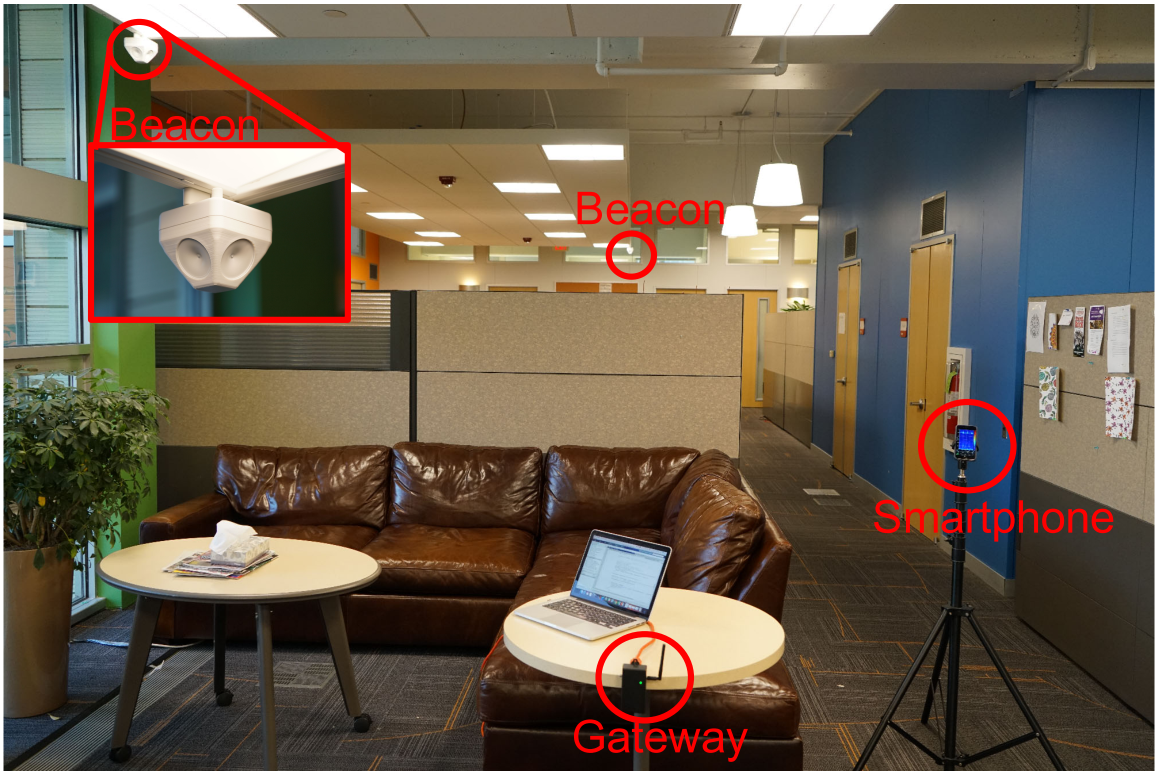


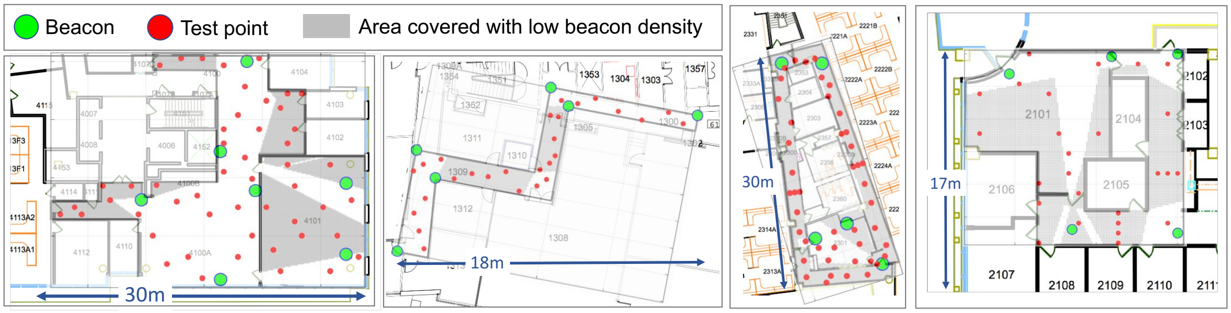
Figure 4.6: Experimental setup for evaluating location acquisition

4.3 Evaluation

In this section, we evaluate the performance (localization accuracy and the accuracy of detecting NLOS) of the floor-plan aware solver in real-world deployments. We also discuss practical aspects related to the proposed approach, such as the trade-off between increasing LOS range and decreasing the amount of NLOS, the number of hypotheses evaluated and the effect of environmental factors.

Description of deployments: The experimental setup is shown in Figure 4.6. The nodes were installed below ceiling tiles and the phone was placed on a tripod. We deployed up to seven ultrasonic nodes in four different environments on Carnegie Mellon University's campus. Figure 4.7 shows the four environments (labeled $F1$-$F4$) with their floor plan, the position of the beacons (green circles) and test locations (red dots).

The modeled floor plan is outlined by solid grey lines, and the regions with low beacon density (one or two beacons) are shaded in grey. The beacon locations were determined manually. We applied the previously presented beacon placement approach to optimize for high coverage, good geometry and low number of beacons. Table 4.3 shows some of the characteristics of the deployments, which inform us about the complexity of the floor plan, beacon coverage density and amount of NLOS. $F1$ includes an open lounge and kitchen area (in the center), a large classroom (on the right) and open areas leading to offices and hallways. Among the floor plans, $F1$ has the maximum number of beacons, leading to the maximum number of feasible hypotheses. It also has a high amount of NLOS (33%). $F2$ has four connected hallways with low geometrical



(a) F1 (b) F2 (b) F3 (b) F4

Figure 4.7: Deployments for evaluating coverage-aware solver

complexity and as few as seven feasible hypotheses. Among the floor plans, this has the highest beacon coverage density, and all regions could be localized without ambiguity in the absence of NLOS. $F3$ has long hallways that are mostly covered by a single beacon and includes a kitchen area which had areas of two or three beacon coverage. Among the floor plans, $F3$ has the least beacon density. Floor plan $F4$ includes a large conference room and open broad hallways around it on two sides.

Solvers evaluated: We evaluate and compare the following solvers:

(1) **GD:** Gradient Descent: This is a common method to estimate location and is most efficient in terms of computation.

(2) **GS:** Grid Search: This solver estimates the minimum mean square error location in a rectangular box surrounding the floor plan. This serves as our baseline solver in the absence of floor plan information.

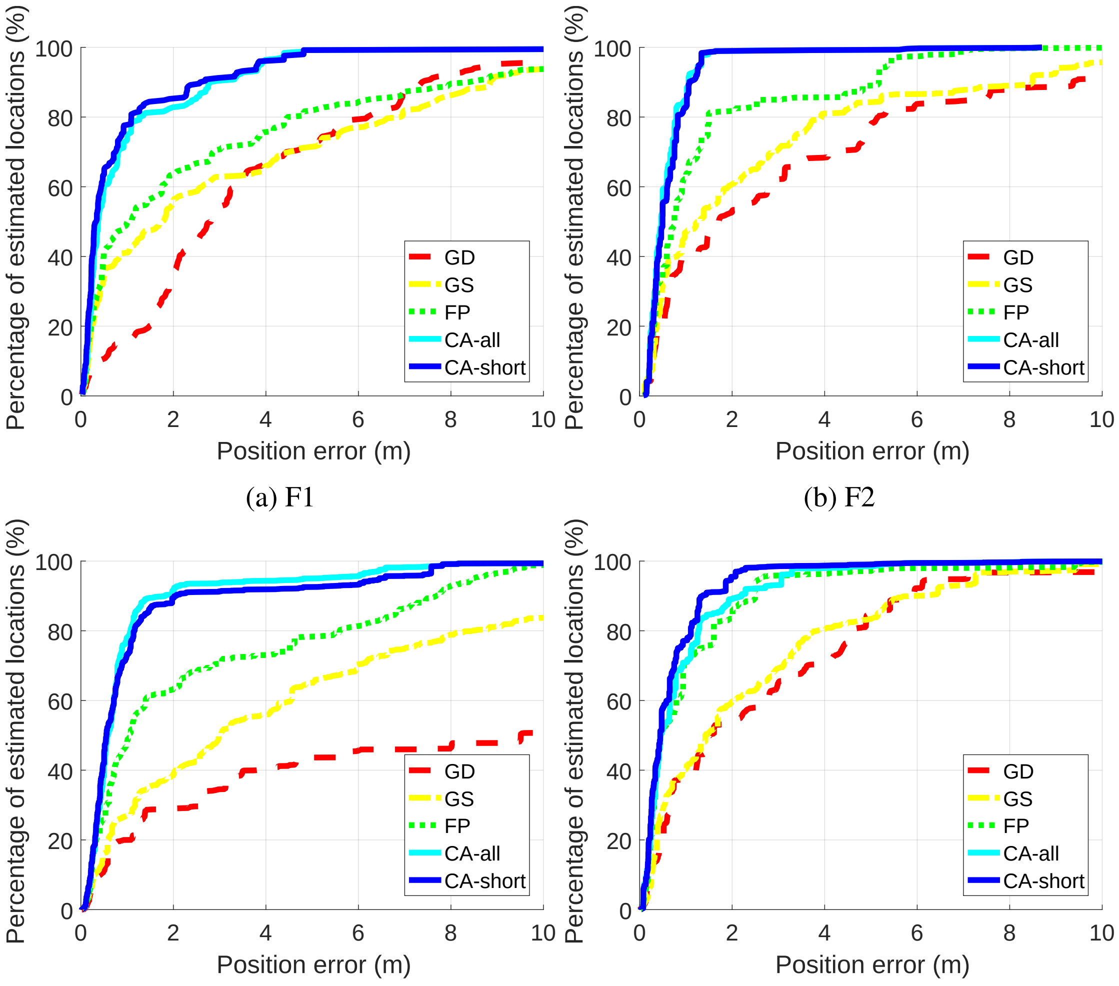
(3) **FP:** Grid Search in Floor Plan: This approach searches for the minimum error solution within the floor plan. This is the baseline for comparing against using the floor plan to constrain the location without using the beacon coverage information.

(4) **CA-all:** Proposed coverage-aware solver which checks for all hypotheses and then selects one.

(5) **CA-short:** Proposed coverage-aware solver that sorts the hypotheses based on the ranges.

Characteristics of real-world deployments					**NLOS detection performance**				
Env	**# vertices**	**# beacons**	**# feasible hyp.**	**% NLOS**	**CA-all**		**CA-short**		
					TP	**TN**	**TP**	**TN**	**# hyp. eval.**
F1	33	7	39	33	0.94	0.86	0.88	0.95	23
F2	10	6	7	24	0.96	0.97	0.82	0.99	2
F3	73	5	19	20	0.83	0.91	0.56	0.98	5
F4	23	5	17	22	0.93	0.88	0.83	0.99	5

Table 4.3: Geometrical parameters of the test environment floor plans and performance of the coverage-aware solver in detecting NLOS, with characteristics of the deployments. (Legend: #: Number of; hyp. eval.:Hypothesis evaluated)



(a) F1

(b) F2

(c) F3

(d) F4

Figure 4.8: Localization performance in real-world deployments

4.3.1 Localization accuracy

Figure 4.8 shows the Cumulative Distribution Function (CDF) of the localization accuracy of each solver across all floor plans. Overall, using the floor plan to confine the solutions (*FP*) improves the performance over not using the floor plan (*GD, GS*), and integrating the coverage information (*CA-all* and *CA-short*) outperforms the other solvers across all deployments. Some key observations are:

(1) *The coverage-aware solver is able to maintain 80th-percentile error of 1m under different environments.* This can be seen in the four CDFs.

(2) *Under high-NLOS conditions, the coverage-aware solver significantly improves performance.* $F1$ has wide open areas with beacons located around corners, resulting in a third of the measurements to be in NLOS, as seen in Table 4.3. We see that the 80-percentile localization accuracy improved from 6m to 1m by integrating the coverage information.

(3) *In low-density deployments, the coverage-aware solver significantly improves performance.* Among the floor plans, *F3* has the lowest beacon density and most areas cannot be localized without ambiguity even under pure-LOS conditions. We see that the 80th-percentile localization performance improved from 8m to 1m with the coverage-aware solver.

(4) *Under low NLOS and sufficient beacon density to localize with LOS without ambiguity, the coverage-solver may not provide much improvement over using only the floor plan.* We observe that $F4$ did not benefit much from the coverage-aware solver as compared to using only the floor plan. This is because it had few NLOS measurements in reality. The classification of 22% of measurements as NLOS is due to inaccuracy in our ray-tracing model by not accounting for diffraction around corners. This floor plan has two thin partitions that we modeled as solid walls; the NLOS measurements through these had low error, the FP solver benefited from using them and the coverage-aware solvers eliminated these measurements.

4.3.2 NLOS/LOS detection accuracy

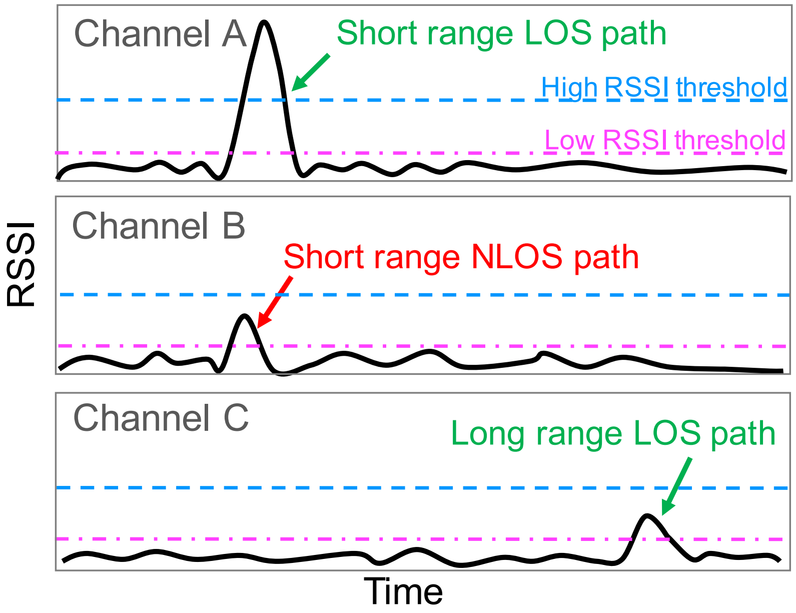
The *TP* and *TN* columns of Table 4.3 show the accuracy of detecting NLOS and LOS correctly. The *CA-all* solver has a *TP* of 91.5% and a *TN* of 90.5% on average. The *CA-short* solver is biased towards a higher *TN* of 97.8% at the expense of a lower *TP* of 77%. This is because the *CA-short* solver checks for the hypotheses in iteration by giving higher weight to hypotheses that have a higher number of LOS beacons. Hence, it is biased towards detecting signals as LOS. We also observe that the performance of *CA-short* is sometimes better than *CA-all*, which is unexpected since *CA-all* evaluates all hypotheses and *CA-short* evaluates only a subset of the hypotheses. This happens when both approaches have selected an incorrect hypothesis. This occurs when (1) there is only one LOS beacon in range located around a corner, such there is a NLOS region closer to the beacon than the LOS location and (2) an NLOS signal from another beacon diffracts around a corner/doorway causing NLOS range error close to zero. In these situations, the *CA-short* is biased towards misclassifying the NLOS as LOS, which provides lower error than *CA-all*, which converges on a consistent hypothesis with close to zero residue by misclassifying the LOS range as NLOS, discarding it and localizing with a single NLOS beacon. These cases tend to be rare in practice, and their occurrence can be reduced with better placement of beacons.

4.3.3 Trade-off between LOS and NLOS performance

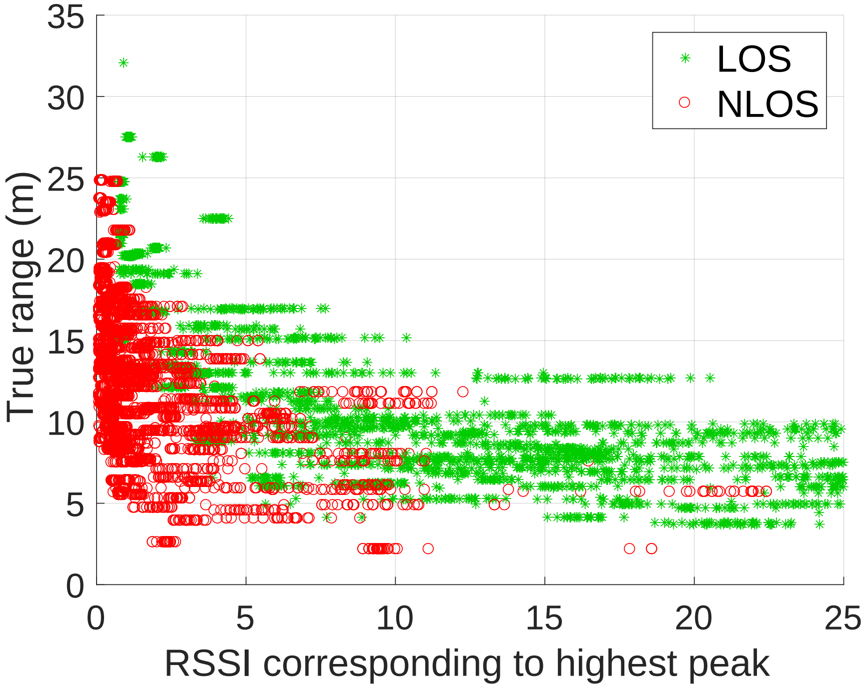
In this section, we discuss the practical trade-off between increasing LOS range and decreasing the amount of NLOS signals. Several range-based systems estimate the time-of-arrival (TOA) in either the received channel impulse response or the result of matched filtering of received signal with transmitted signal. The TOA is estimated by selecting a peak in the received signal based on the RSSI using algorithms such as first peak above noise floor, highest peak above noise floor, first peak within a window before the highest peak, etc. A detailed survey of the approaches can be found in [37, 74, 118]. However, a persistent challenge is to determine the threshold of RSSI for peak detection. This is illustrated in Figure 4.9a with three types of channels. Channel A represents a short range LOS path. Channel B represents an NLOS path of the same range. Channel C represents a long range LOS path and has comparable signal strength to Channel B. If a high threshold for peak detection is selected, the NLOS signal in Channel B and the long range LOS signal in Channel C would not be detected. On the other hand, if a low RSSI threshold is selected, we would detect both signals. Hence there is a trade-off between the amount of NLOS signals detected and the maximum range of LOS signals that are detectable. Figure 4.9b shows this trade-off from real-world data collected using our beacon platform in several environments. If the threshold is set to be high, we can eliminate most of the NLOS, but the LOS range is limited. Since our localization system is robust to NLOS signals under the assumption that NLOS signals are positively biased, we set the threshold to a low value of 1 and allow NLOS signals in order to have a long range of 25m for LOS. Allowing for higher LOS range reduces beacon density as it allows us to cover a larger area with fewer beacons.

4.3.4 Complexity and number of hypotheses

We first make a distinction between the number of hypotheses feasible for a deployment and the number of hypotheses evaluated while solving for location. The number of feasible hypotheses for a floor plan grows with the number of beacons and is highly dependent on the geometry of the floor plan. If we consider all combinations of N beacons, we can get a maximum of 2^N combinations of hypotheses. The number of feasible hypotheses is much lower due to the structured nature of floor plans. As a comparison, the floor plan in Figure 4.3 has five beacons, and hence 32 possible hypotheses, but the number of feasible hypotheses for the floor plan is six. For large, building-scale floor plans, the number of feasible hypotheses can be high, but among the feasible hypotheses, only the hypotheses containing one of the beacons in range to be in LOS will be evaluated. The last column of Table 4.3 shows the number of hypotheses evaluated on average across all test points for *CA-short* before the algorithm converged. If we notice the difference between this and the total number of hypotheses feasible for this floor plan (fourth column), we see that the *CA-short* approach evaluates fewer hypotheses. The cost of



(a) On the similarity of a long-range LOS signals and a short-range NLOS signal



(b) Variation in RSSI with range for LOS and NLOS signals

Figure 4.9: Trade-off between increasing the range of detectable LOS signals and detecting a higher amount of NLOS signals

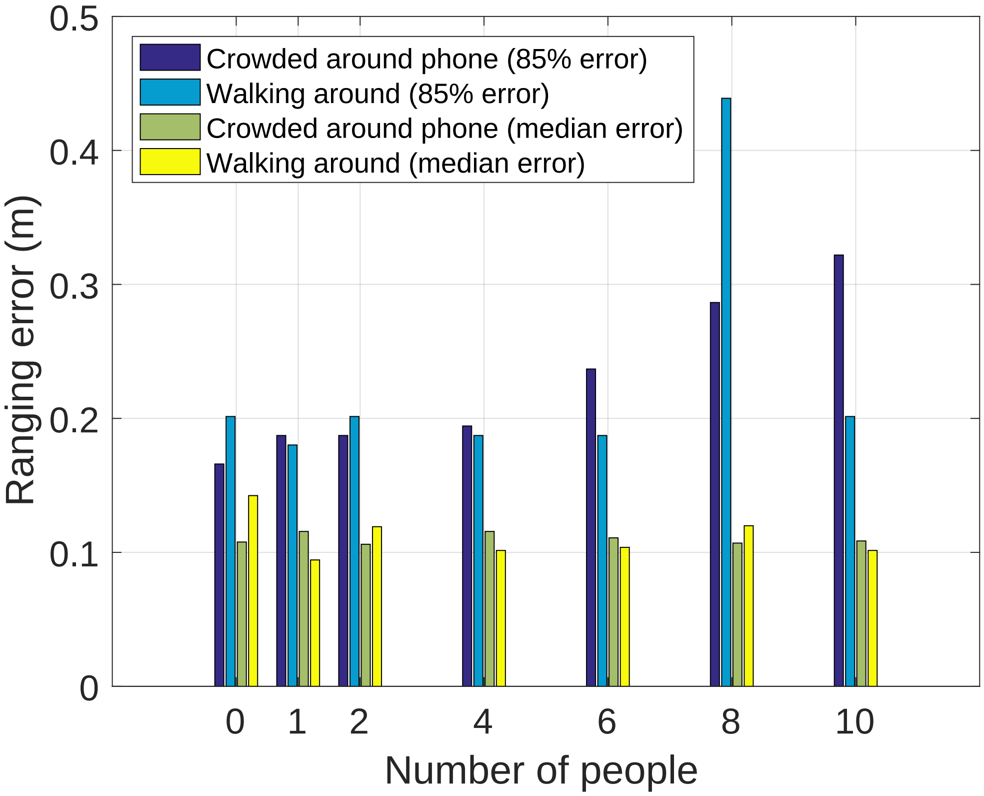


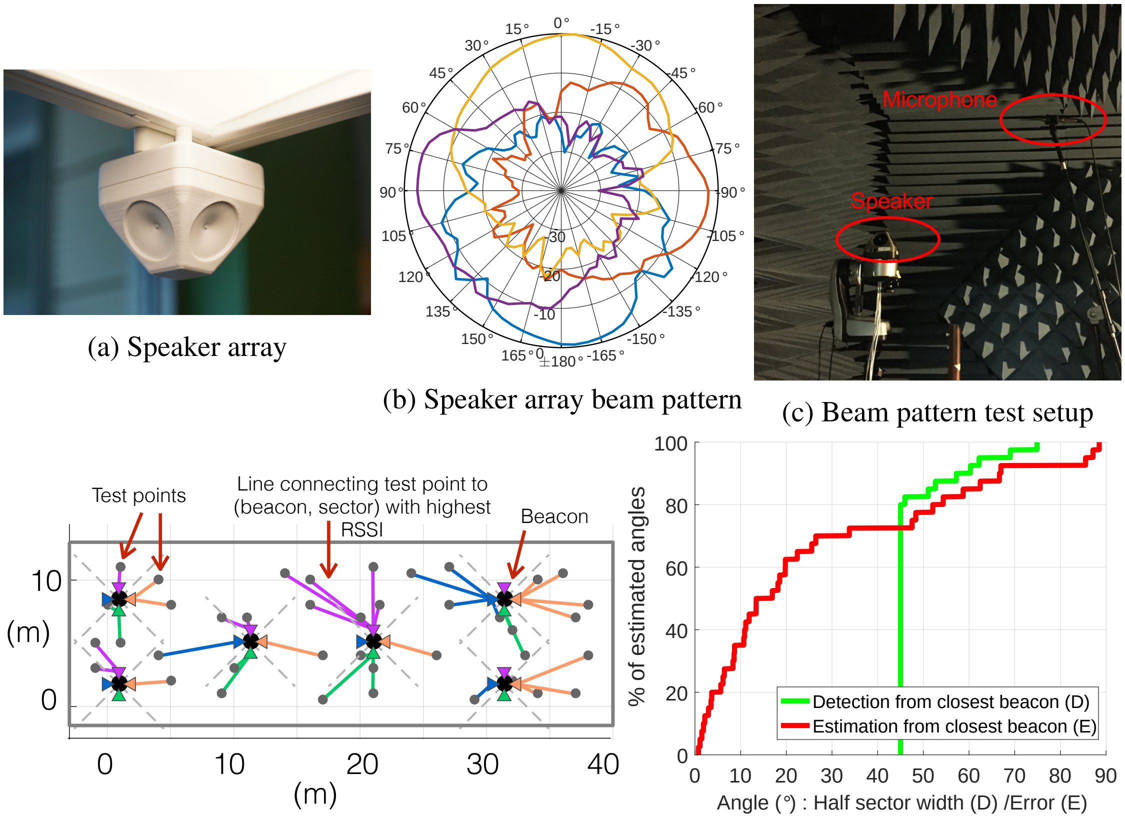
Figure 4.10: Impact of people's presence on received ranges

considering a hypothesis is the cost of computing the mean square error across all locations in that hypothesis. The complexity at worst compares with performing a grid search (min mean square error estimation) which in practice works well even on large floor plans.

4.3.5 Environmental effects

Our ray-tracing model assumes that the NLOS signals are caused by signals reflecting off walls. Two environmental factors that challenge this assumption are the presence of doors (doors can be open or closed but the ray-tracing model is deterministic) and the presence of people (people move around and reflect/obstruct signals). In this section, we discuss the impact of these factors.

Effect of doors: For the system to work in a region when a door is closed, a sufficient number of beacons should be deployed to provide coverage assuming the scenario of the door being closed irrespective of the localization approach. During ray-tracing, we model the path between a location and a beacon across a doorway as LOS. If the door is closed, the true hypothesis is penalized for having a missing LOS range. However, this hypothesis would still rank higher than the others, since sufficient LOS beacons would be in range even without the range from the beacon across the doorway. When the door is open, we receive a LOS range from the beacon which is utilized for localization and the hypothesis assuming this is a LOS beacon will be selected. If a door is generally left open when the environment is in use, we recommend assuming the doorway is always open in order to deploy fewer beacons.



(a) Speaker array

(b) Speaker array beam pattern

(c) Beam pattern test setup

(d) Real world deployment to characterize sectors

(e) Accuracy of angle estimation and sector detection

Figure 4.11: Leveraging ultrasonic sectored speakers for improving coverage information

Effect of people: To study the impact of people in the environment on the received signals from the beacons, we performed an experiment in the environment shown in Figure 4.6 with the help of 10 participants. A stationary person would hold the test device in hand. In the first test scenario, the participants were crowded around the test device, either standing still or walking, within a 0.5m radius. In the second test scenario, the participants were walking around the test device within a 4m radius. The error in range measurements from four beacons deployed in the test area is shown in Figure 4.10. We have shown the median error and the 85 percentile error. When people are walking around, there is little impact because the beacons are deployed at ceiling level and people are unlikely to block the signals unless they are very close to the device. The performance degrades when people are crowded around the phone. Though the ranging error increases due to the diffraction of sound around people, they do not completely block the direct path and the errors are less pronounced compared to wall obstructions.

4.3.6 Improving localization accuracy with sectored transmitters

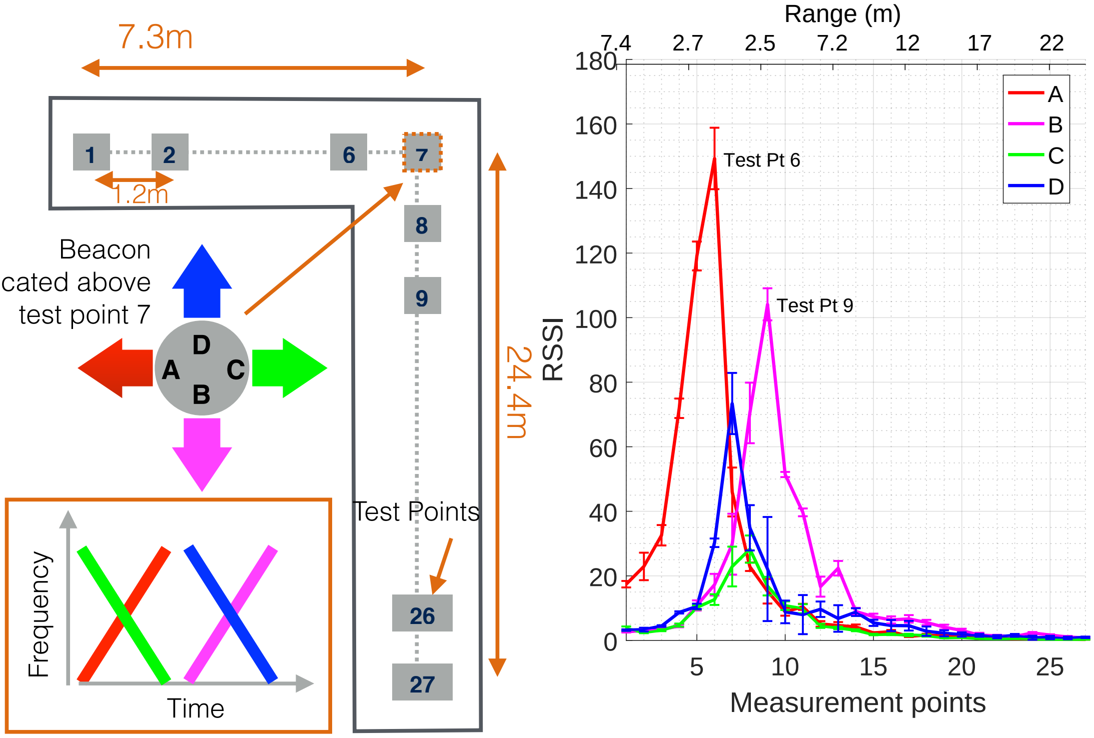
So far, we have assumed that the beacon coverage is omnidirectional and confined by the floor plan. In this section, we discuss how coarse angle information from a beacon can improve the localization in extremely low beacon density environments. The intuition is that the coarse angle provides finer grained coverage information than an omnidirectional beacon transmitter and this coverage information can be incorporated into the location solver. Though this section is specific to the design of our ultrasonic beacon platform (introduced in Section 2.1), the concepts can be applied to other beacons with angular information. We are able to obtain coarse angle information by using a circular array of speakers at a beacon. Each speaker has a direction beam pattern approximately confined to a cone. We use a sectored array to solve two problems: (1) we can more easily generate uniform ultrasonic coverage and (2) we can accurately estimate the coarse direction to the beacon just based on signal strength metrics.

Figure 4.11b shows the horizontal beam pattern of one of our speakers (Figure 4.11a) obtained from testing in an anechoic chamber (Figure 4.11c). The beam pattern shows the receiver correlation magnitude of the ultrasonic chirps transmitted at various angles in the horizontal plane to the microphone. This particular speaker was selected for its wide beam patterns across our desired frequency range supporting approximately a 90-degree sector. Upon testing, we realized that transmitting the same signal causes interference (combing) between adjacent speakers, which significantly distorts the signal at boundaries. This indicated that we needed to orthogonally code or time multiplex neighboring speakers to avoid potential interference. For this reason, selecting four sectors in our array provides the best uniform coverage with enough angular resolution to enable hypothesis pruning while maximizing update rate.

There are multiple ways to use the individual speakers' RSSI for inferring sectors or angles. For selecting or eliminating among multiple hypotheses, we can use the speaker with highest RSSI. Figure 4.11d evaluates the performance of detecting the sectors in a real-world deployment with six beacons. The test points are marked in grey, and are connected to the sector-beacon pair with the highest RSSI. We see that in general it selects the closest sector of the beacon. We observe that due to variation in RSSI with angle of sector, it does not always select the closest beacon. If we assume a $90°$ sector beam pattern and detect the sector based on the speaker with highest RSSI, the performance of the sector detection scheme is shown by the green D line in Figure 4.11e. When tested uniformly across all angles, we accurately selected the correct $90°$ sector 80% of the time.

The second approach is to compare RSSI from the two dominant speakers to estimate the angle with respect to the beacon. Here, we assume that all speakers have an identical beam pattern with a maximum width of $180°$. Under this assumption, at any angle we ideally should receive signals from only two speakers, and the sum of amplitudes from the two speakers should be constant at all angles. To translate this model to practice, we only consider the RSSI from the two dominant speakers. The accuracy of angle estimation is shown by the red E line in Figure 4.11e. Using RSSI from the two dominant speakers, we were able to estimate the angle within $90°$ for 72% of the test measurements and to within $120°$ of the center of the correct sector 84% of the test measurements.

Another approach to incorporate the sector information into the solver without altering the localization algorithm is to partition each beacon's coverage further into disjoint zones based on the individual sectors. Though the total number of feasible hypotheses increases, since the sectors of a beacon are disjoint, a much smaller region of the floor plan will be searched while estimating the location. Subsequently, we only need to consider the hypothesis corresponding to the LOS coverage of that sector. Finally, we also use the sector purely for disambiguating between possible locations. However, while localizing, we first detect the sector based on the



(a) Experimental setup

(b) RSSI of the four sectored speakers at different locations

Figure 4.12: Localization using single beacon with sector-based coverage

highest RSSI among the four speakers. The benefit of sectors can be seen in deployments much sparser than the deployments we have shown. For instance, small rooms or long hallways would only require a single beacon placed in the center of the room. Figure 4.12 shows an example where the sectors can help in determining which hallway the mobile device is in, hence a single beacon is able to localize two hallways, rather than requiring three beacons in each hallway. It also shows the chirps transmitted by each of the sectors at the bottom left.

4.4 Summary

In this chapter, we presented a solver that localizes robustly in the presence of low beacon density and high NLOS signals by integrating the floor plan geometry and beacon coverage models. The proposed solver shifts the 80% accuracy point from $48m$ to $1m$ as compared to solvers that do not use the floor plan information, for the same beacon density. We are able to detect and remove NLOS signals with 91.5% accuracy without requiring additional infrastructure.

Chapter 5

Mapping

In Chapter 3, we addressed the question of where to place beacons given a floor plan, and in Chapter 4, we discussed how to solve for locations given range from beacons, beacon locations and the floor plan. In this chapter, we cover the creation of beacon maps and maps of the floor plan. Key to our mapping approach is that we leverage mobility of devices over time, which effectively provides us additional sensor information that we can use for mapping. In Chapter 6, we build on the approaches presented in this chapter to build maps of spatially varying signals, specifically the magnetic field, that we use subsequently for acquiring the device orientation.

First, we elaborate on the difference between beacon placement, described in Chapter 3, and beacon mapping, which we present in this chapter. Beacon placement strategies determine locations at which beacons should be placed. However, we might not always have control over placing beacons. In the future, we envision that WiFi access points, IoT devices and other smart devices with ranging capabilities would act as beacons. In these scenarios, devices that are deployed in the environment are opportunistically used as beacons, and we have to estimate their locations; this is the beacon mapping process. As seen, beacon mapping is complementary to beacon placement. In real-world deployments, either one of the problems could be applicable or they both can coexist. For example, some beacons are strategically placed, and the opportunistic beacons that find their way into the localization environment are mapped. Here, the beacons placed beforehand at known locations can improve the mapping accuracy of the opportunistic beacons. Another key difference between the beacon placement and beacon mapping problems is the role of floor plans. In the beacon placement problem, we place beacons given a floor plan. In the beacon mapping problem, on the other hand, we can treat the floor plan map and the beacon map independently. For range-based localization systems, though the floor plan maps are not necessary, knowledge of the floor plan can make the localization robust, as elaborated on in Chapter 4. Finally, whether a system designer adopts a beacon placement or a beacon mapping process is also a matter of cost. The beacon placement problem is motivated by placing the fewest beacons required. If beacon cost is not a major constraint, then one can opt to first deploy beacons at convenient locations and then automatically map them.

Mapping beacons and the floor plan is typically a labor-intensive process or requires expensive robotics systems. In contrast, we solve the mapping problem by using data from pedestrian-held mobile devices that range to beacons. We track the locations of the mobile devices by fusing data from inertial sensors, cameras and beacons.

In Section 5.1, we discuss related work in beacon mapping and floor plan mapping. Our mapping techniques are built on approaches for tracking the mobile device accurately using sensor fusion. First, we present two approaches for tracking mobile devices. In Section 5.2, we present the first approach to tracking: fusion of beacon ranges and VIO using a Particle Filter. Next, in Section 5.3, we present the second approach to tracking: fusion of beacon-based location estimates with pedometer data using an Extended Kalman Filter. Given these two approaches, we tend to use the first when we have low beacon density and when the camera data is available for fusion, and tend to use the second when we have a sufficiently high beacon density and cannot rely on the camera data for VIO. In Section 5.4, we build on the PF-based fusion and present a beacon mapping technique that we implement and evaluate with UWB beacons. In Section 5.5, we present a floor-plan mapping technique that we implement and evaluate with ultrasonic beacons.

5.1 Background and related work

In this section, we cover the background and related work in the area of mapping beacons and indoor floor plans.

5.1.1 Beacon mapping

Several approaches exist for mapping range-based beacons. The suitability of a mapping approach is determined by the inter-beacon ranging capability and the density of beacons. If the beacons range to each other and form a globally rigid graph, then the beacon ranges alone are enough to uniquely map the beacons with respect to each other. The requirements for a topology to be rigid are well understood [14, 38, 83], and several techniques exist to solve for the beacon locations under a rigid topology [15, 19, 44, 51, 83, 112]. However, if the network is not globally rigid, the same inter-beacon ranges can produce multiple realizations of the beacon locations. Further, in many cases, we may not have inter-beacon ranges. In these cases, we use mobile devices, and fuse the range between the devices and the beacons and inertial sensors of the device to map the beacons. In real-world indoor deployments, due to the limited range of beacons and the signal attenuation through walls, we expect this to be the case rather than having a fully connected rigid network. Hence, to solve the mapping problem, we leverage mobile devices in the environment that range to beacons. Effectively, mobility creates multiple instances of a device over time in different locations and provides more sensor information than is available from a single snapshot of time.

The problem of simultaneously localizing a mobile device and mapping range-only beacons is referred to as Range-Only SLAM (RO-SLAM). There are many solutions in the robotics literature for RO-SLAM. The algorithms include approaches based on Extended Kalman Filter-SLAM [15, 19, 34, 55, 64, 86, 89], GraphSLAM [52], Spectral Learning algorithms and Particle Filters [15, 64, 86]. RO-SLAM is challenging since the problem is under-defined, i.e., we only have the distance between beacons and a target device. This leads to multiple feasible hypotheses for beacon locations, and as a result, typical linear and Gaussian approximations of measurement models do not fit the measurements. To solve this problem, we adopt the Rao-Blackwellized

Particle Filter (RBPF) approach [35, 131]. The RBPF uses particles to represent the posterior over the mobile target location, along with other parametric probability distribution functions to represent the beacon locations. While the probability distribution of the mobile device is represented by particles, the probability distribution of the beacons can be represented by an Extended Kalman Filter [33, 82, 124], a Sum of Gaussian-based EKF [21, 24, 39] or Particle Filters [22]. In specific, we adopted the algorithm proposed in [22] where both beacons and the mobile device are represented as particles. This gives us the flexibility to cope with NLOS measurements, initialize beacons at different times and represent the probability distribution of beacons more accurately than EKFs since we do not have to fit the distribution to a Gaussian. Our work differs from these systems in robotics as we implement and evaluate indoor ranging-beacon SLAM captured by a pedestrian walking around with a hand-held mobile phone, rather than a robotic system.

5.1.2 Mapping floor plans

Often we may not have the floor plans from buildings where we wish to deploy a localization system. Even if we have the floor plans, it is possible that they are outdated. Floor plan reconstruction techniques leverage acoustic, laser scanning, LIDAR and camera sensing systems. Acoustic-based systems that reconstruct floor plans by estimating the location of walls from echoes [106, 116, 129] are promising given that smart speakers are becoming common in indoor environments. However, these approaches have mostly been shown for single room deployments and require the speaker and microphone setup to be moved to all regions of the building to truly reconstruct the floor plans. Robots mounted with 3D laser range finders or LIDAR systems [125, 130] are often used to map commercial spaces. These systems produce 3D point clouds, from which we can extract the 2D floor plan, as shown in [88]. A more detailed survey of mapping algorithms can be found in [132]. Recently, camera-based mapping of spaces has been shown to be feasible [47, 126, 143] and is promising given that pedestrians can scan environments with their mobile devices to create the maps. Google's project Tango and sensors like Occipital's Structure use depth sensors to scan and map 3D environments.

Our work augments these techniques, but also differs from them, as we present a pedestrian-guided mapping process where a pedestrian simply walks around and touches the corners of the indoor environment that they wish to be part of the map. This does not require any expertise or expensive instrumentation. Rather than using an automated system that detects all obstacles in the environment, in our approach, the system installer can decide which parts of the environment to include in the map. For instance, for acoustic beacons deployed at the ceiling level, if there are no tall obstacles such as cubicle partitions that block the beacons from users, then it is sufficient to map only the corners of the rooms. The simplicity of our mapping process makes it easy for users to quickly re-map the environment if it changes or if new barriers are introduced.

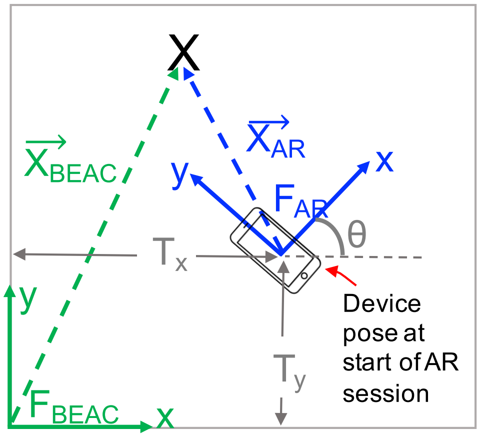

X
$\overrightarrow{X}_{BEAC}$
$\overrightarrow{X}_{AR}$
y
x
F_{AR}
θ
T_x
y
T_y
Device pose at start of AR session
F_{BEAC}
x

Figure 5.1: Estimation problem for beacon ranges and VIO fusion. Beacon ranges are measured in reference frame F_{BEAC} and VIO is measured in reference frame F_{AR}. We are required to estimate the transformation between the reference frames, i.e., T_x, T_y, θ.

5.2 Tracking approach 1: Fusion of beacon ranges and Visual Inertial Odometry using a Particle Filter

In this section, we present an approach for precise tracking of a mobile device in a frame of reference fixed and defined by the beacons. We fuse asynchronous ranges from beacons (measured in the frame of reference defined by the beacon locations, F_{BEAC}) and VIO data (measured in an AR reference frame, F_{AR}) using a Particle Filter. Range from beacons provides absolute location estimates and VIO provides tracking at a high accuracy and high update-rate. We fuse both these sensor inputs to accurately track the six-degrees-of-freedom pose of the mobile device in the beacon reference frame [105].

The notations we use are:

- F_{BEAC}: Beacon frame of reference that is fixed over time across users.
- F_{AR}: AR frame of reference that is fixed for an AR session on a single mobile device.
- X_{AR} : 3D VIO position, i.e., x,y,z values of the device in F_{AR}.
- X_{BEAC} : 3D position of the device in F_{BEAC}.

We show the 2D (x-y) view of these reference frames in Figure 5.1. We use F_{BEAC} as the external fixed frame in which we represent all maps and locations. The device is tracked by VIO in an AR frame of reference, F_{AR}. A location X_{BEAC} in the beacon coordinate frame is represented as X_{AR} in the AR coordinate frame. For the estimation problem, we have to find the six-degrees-of-freedom transformation between the beacon frame F_{BEAC} to F_{AR}, or equivalently between X_{BEAC} and X_{AR}. F_{AR} and F_{BEAC} have one axis parallel to the gravity vector. Though this is provided to us by ARKit, if that is not the case, we can compute the gravity vector by filtering the accelerometer data when the phone motion is low [146]. Having this axis aligned, we reduce the six-degrees-of-freedom frame conversion problem to a three-degrees-of-freedom translation and a one-degree-of-freedom rotation. Figure 5.1 shows the translation in 3D and the rotation.

$$X_{BEAC} = R(\theta_{AR\rightarrow BEAC}).X_{AR} + T_{AR\rightarrow BEAC} \tag{5.1}$$

where

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

and

$$T_{AR\rightarrow BEAC} = [T_x, T_y, T_z]^T \tag{5.3}$$

Ideally, if there is no drift in VIO and no noise in VIO or range measurements, $(T,\theta)^{AR\rightarrow BEAC}$ would be time-invariant. However, in reality, the transformation is time-varying, and for accurate rendering we have to continuously update $(T,\theta)^{AR\rightarrow BEAC}(t)$. The Particle Filter fusion algorithm estimates $X_{BEAC}(t)$ and $\theta(t)$. We read X_{AR} from the VIO API and use these estimated quantities with Equation 5.1 to compute $(T,\theta)^{AR\rightarrow BEAC}$.

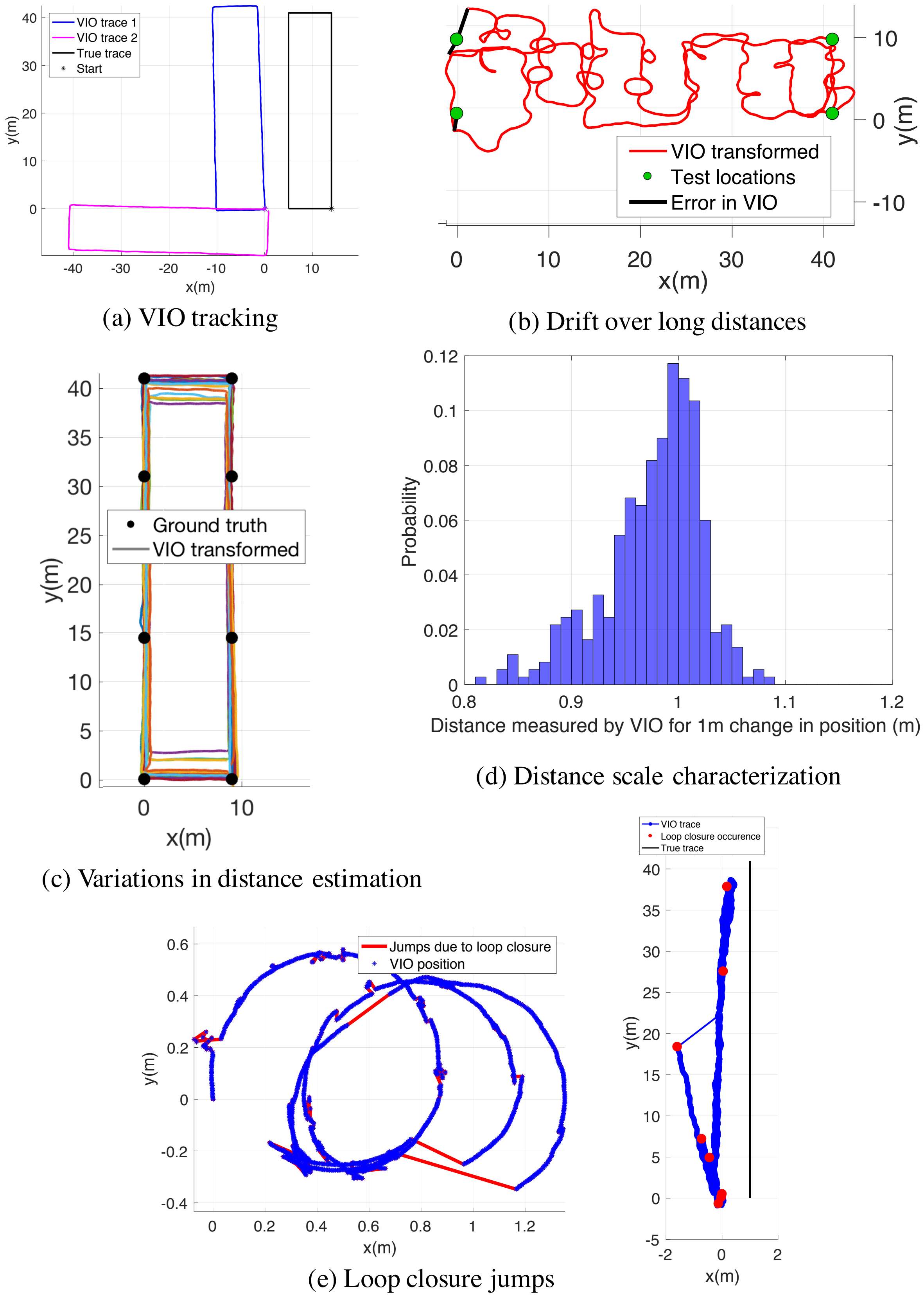
Next, we characterize the VIO tracking in Section 5.2.1 and describe the sensor fusion in Section 5.2.2.

5.2.1 ARKit Visual Inertial Odometry

In this section, we characterize the tracking performance of VIO from Apple's ARKit framework. We evaluated ARKit 1 with iPhone 8 and iPad Pro 2018 devices. At the time of testing, this was the best performing VIO available on commodity devices. Motion tracking is performed by fusing tracked visual feature points with inertial sensor motion data. In most implementations of VIO, relocalization or loop closure is performed, where in addition to tracking, the location is corrected when previously acquired visual features in the environment are revisited. The output of VIO is the position and orientation of the device with respect to the AR reference frame at startup. Below, we characterize ARKit tracking and then characterize the VIO performance.

Frames of reference

ARKit provides three options for setting the coordinate system using the *WorldAlignment* parameter in *ARConfiguration* [5], and in all cases, the coordinate system is right-handed. (1) *Camera*: the frame of reference is fixed with respect to camera, i.e., it moves with the device; (2) *Gravity*: the origin is the location of the device upon startup of session. The +y axis is parallel to gravity pointing up and the -z axis points in the direction in which the rear camera points upon startup; (3) *Gravity And Heading*: this is similar to *gravity* except that the -z axis points towards true north, as measured by the device upon startup. In reality, the magnetic field inside buildings varies drastically, and this frame is typically only useful outdoors, but not indoors. ARCore follows a similar convention. In our work, we use the *Gravity* configuration with the (z, x) coordinates of ARKit for horizontal positioning and the y-axis for vertical positioning (to keep the vertical axis aligned with gravity and right-handedness). Figure 5.2(a) shows the VIO positions of the device in the horizontal plane for two different AR sessions. The true trace in world frame is shown in black. We see that the VIO trace has the same shape as the true trace but starts at (0,0) and has an arbitrary rotation with respect to the true trace.



(a) VIO tracking

(b) Drift over long distances

(c) Variations in distance estimation

(d) Distance scale characterization

(e) Loop closure jumps

Figure 5.2: Characterization of ARKit tracking

VIO tracking performance

To characterize the tracking performance of ARKit, we marked ground truth test points with known locations on the floor in multiple environments. Multiple users walked traces connecting the test points with a device held in hand. Below, we characterize key parameters of VIO tracking.

Update rate: The update rate of VIO is a function of the frame rate of the camera and the sampling rate of the sensors. The highest update rate on current devices is $60Hz$ [1, 2].

Drift: Figure 5.2(b) shows a $200m$ trace of VIO, where the overall drift is $5m$. The drift is due to cumulative errors in the angle estimates, which are higher when the user makes turns while walking. To account for this drift, we assume the error in angle between two position updates separated by time δt is drawn from $\mathcal{N}(0, \delta t \cdot \sigma_\theta^2)$ where σ_θ is empirically evaluated.

Distance estimation: Figure 5.2(c) shows the VIO tracking over a rectangular trace of perimeter $100m$ in an office environment. We observe that the VIO traces are scaled to different extents as compared to the ground truth trace (shown by the black dots). This error in scale is due to incorrect distance estimation, which often occurs when a single camera is used and depth cannot be estimated correctly. Across all the experiments we conducted in real-world settings with trace lengths ranging from $10m$ to $300m$, we estimated the error in VIO distance estimation, shown in Figure 5.2(d). This data implies that when the position changes by δd, the distance as estimated by VIO is drawn from a distribution $\mathcal{N}(0.98\delta d, (0.05\delta d)^2)$.

Errors in relocalization: Figure 5.2(e) shows two scenarios where there are abrupt jumps in position due to incorrect relocalization or loop closure. In the scenario on the left, we placed a phone on a cart and rotated the cart without any translation movement. In the scenario on the right, we walked through a hallway facing identical cubicles in periodic intervals and saw jumps due to different physical locations being visually identical. However, we do not observe abrupt jumps in longer traces, since there is a limited time-memory for the features, and relocalization requires the features in memory to match currently visible features. This also highlights some of the challenges and limitations of heavily vision-based localization.

Although the errors in VIO are specific to the hardware and algorithms on-device, we assume that VIO provides sufficient relative positioning to support AR applications.

5.2.2 Sensor fusion

We chose a Particle Filter (PF) approach for our state estimation for the following reasons: it (1) can run in real-time on phones and only requires the previous state of the filter, (2) allows us to accommodate for arbitrary noise models used to describe VIO and beacon range errors, (3) can continue to work in under-defined cases with as few as one to two beacons and (4) is agnostic to update rate and allows us to handle asynchronous ranges from the beacons without requiring us to receive synchronous ranges and perform trilateration.

The state of the device at time t is represented by N 5-dimensional particles (x, y, z, θ, s), with weights w_t^i, $i \in [1, N]$, where (x, y, z) is the 3D position of the device in F_{BEAC}, θ is the rotation of AR frame with respect to world frame and s is the scaling factor of AR distance as compared to true distance. The intuition behind this formulation is that the particles with θ closer to the true value will be updated to the true locations when VIO provides motion updates, and subsequently they will be weighed higher with ranges from beacons since they will be at the correct location and hence consistent with the range measurements.

The general PF algorithm has four functions: initialization, motion model update, measurement update and resampling [131], which we describe below in context of our approach:

Initialization:

We initialize the particles based on the first few range measurements to overcome poor initialization in case NLOS range measurements are received during initialization. The five states (x, y, z, θ, s) are initialized as:

- Draw z from nominal user height $z \sim \mathcal{U}(0.7, 1.3)$.

  If we assume arbitrary starting height, we would have to initialize particles on a sphere. This makes it challenging to run the PF in real-time on the current devices, since our state estimation has five dimensions. So our implementation initializes the z state uniformly distributed about a nominal holding height of 1m. The particles at the incorrect (x, y, z) will be weighed lower when we receive additional beacon ranges.
- Draw $[x, y]$ from measurement model of beacon ranges, the 3D location of the beacon, the range measurement from the beacon and z.
- Draw s from $\mathcal{N}(0.98, (0.05)^2)$, based on Distance Estimation characterization in Section 5.2.1.
- Draw θ from $p(\theta|x, y)$, based on the magnetic field map. When the map has no measurement at (x, y), θ is drawn with uniform probability in $[0, 2\pi]$. We describe the magnetic map in detail in Chapter 6.

Update of states based on motion model:

The input u_t to the Particle Filter is the change in VIO position of the device. $u_t = [\Delta x, \Delta y, \Delta z]^T$ where $[\Delta x, \Delta y, \Delta z] = X_{AR}(t) - X_{AR}(t-1)$. We assign Σ_t^{xyz}, the covariance matrix corresponding to noise in VIO position, to be proportional to the distance change, with a constant σ_{xyz} determined empirically, where $\Sigma_t^{xyz} = I_{3\times3}.(\Delta x_t^2 + \Delta y_t^2 + \Delta z_t^2).\sigma_{xyz}^2$ Making the covariance value proportional to the change in distance enables us to recover from scenarios where there is an incorrect loop closure in VIO tracking. In this scenario, the change in distance during loop closure is high, increasing the uncertainty in the location of the particles after the VIO update. Every VIO input updates the state of the particles as:

- Draw $[x, y, z]_t$ from $\mathcal{N}([x, y, z]_{t-1}^T + s_{t-1} \cdot R(\theta_{t-1}) \cdot u_t, \Sigma_t^{xyz})$ where $R(\theta)$ is defined in equation 5.1. Each particle gets updated in a direction, based on its belief of θ. Eventually the particles with θ close to the true value will be weighed higher, as they will be closer to the true location and will be weighed higher when a range measurement is received.
- Draw θ from $\mathcal{N}(\theta_{t-1}, \sigma_\theta^2)$, to account for the error as described in the ARKit drift characterization.
- Draw s from $\mathcal{N}(s_{t-1}, (\Delta x_t^2 + \Delta y_t^2 + \Delta z_t^2).\sigma_s^2)$

Update of weights based on measurement model:

The measurement to the filter is a range from a single beacon. When a range update is received, we update the weight of the particles w_t^i from the measurement model. The measurement model is specific to the beacon technology and ranging error model.

Resampling:
We perform resampling if the effective sample size is < 0.5 and add some noise to perturb the state of particles to introduce diversity as described in [131].
Finally, we estimate the state of the filter by taking the weighted mean of all the particles and estimating the uncertainty in location from the weighted standard deviation of all particles.

In summary, the states are updated from VIO measurement X_{AR}, and the weights are updated from the range measurement. During initialization, the location (x, y, z) is initialized from the range measurement. If the magnetic field map is available, the angle θ is initialized from the map. Otherwise, the angle is initialized by drawing particles uniformly from $[0, 2\pi]$. With motion inputs, each particle gets updated in the direction indicated by its belief of $\theta_{AR\rightarrow BEAC}$. Eventually, particles with $\theta_{AR\rightarrow BEAC}$ close to the true value will be closer to the true location and hence weighed higher when a range measurement is received. As per Equation 5.1, we estimate $T_{AR\rightarrow BEAC}$ from $\theta_{AR\rightarrow BEAC}$, X_{BEAC} which we get from the states x, y, and from X_{AR}, which is the location of the device in F_{AR} gotten by reading the VIO data.

5.3 Tracking approach 2: Fusion of beacon-based location estimates and pedometer data using an Extended Kalman Filter

When we have a high density of beacons and do not have access to visual inertial odometry data, we implement an Extended Kalman Filter to fuse location estimates from beacons with pedometer and heading data from inertial sensors to track the location of the user. In our implementation of the EKF, we localize the device using beacon ranges and filter the location estimates of a mobile user by utilizing the phone's IMU sensors for tracking. We implemented and evaluated this with ultrasonic ranging beacons and an iPhone as the mobile device receiver [66]. We use the step count from the iPhone's accelerometer and the direction from the compass, which already fuses the magnetometer with the rate gyros. The details of our process model and measurement model for the EKF are given below.

Our objective is to estimate the 2D position (x_t, y_t) of the mobile device at time t. We define the state vector as:

$$X_t = \begin{bmatrix} x_t \\ y_t \end{bmatrix} \sim \mathcal{N}(\mu_t, \Sigma_t)$$

where μ_t is the expected value of X_t and Σ_t is the uncertainty in the state. The EKF generates estimates of μ_t and Σ_t based on the prediction from the previous state X_{t-1} and the process model, and then updates this estimate based on measurement Z_t and the measurement model. A time step of $t = 1$ is the time a person takes for one step while walking.

5.3.1 Process model

The input u_t to this system is given by:

$$u_t = \begin{bmatrix} \Delta D_t \\ \theta_t \end{bmatrix}$$

with noise v_t such that:

$$\begin{aligned} v_t &= \begin{bmatrix} v_t^D \\ v_t^\theta \end{bmatrix} \sim \mathcal{N}(0, M_t) \\ M_t &= \begin{bmatrix} \sigma_D^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix} \end{aligned}$$

ΔD_t is the step length of the mobile device and θ_t is the heading. The step length and heading of the mobile device can be estimated from its IMU sensors and are used as input to the filter. σ_D^2 and σ_θ^2 are the variance in the step length and heading respectively. The focus of our work is not on implementing an accurate step length and heading estimation method, so for our model with the ultrasonic platform we conservatively assumed that $2\sigma_D$ is $10cm$ and $2\sigma_\theta$ is 45°. (For a normal distribution, 95.45% of the values lies within 2σ of the mean.)

The process model is given by

$$\begin{bmatrix} x_t \\ y_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \end{bmatrix} + \begin{bmatrix} (\Delta D_t + v_t^D)\cos(\theta_t + v_t^\theta) \\ (\Delta D_t + v_t^D)\sin(\theta_t + v_t^\theta) \end{bmatrix}$$

The process model is linearized and μ_t and Σ_t are updated as:

$$\begin{aligned} \mu_t &= g(\mu_{t-1}, u_t) \\ \Sigma_t &= G_t \Sigma_{t-1} G_t^T + R_t \end{aligned}$$

where

$$\begin{aligned} g(\mu_{t-1}, u_t) &= G_t \mu_{t-1} + \begin{bmatrix} \Delta D_t \cos(\theta_t) \\ \Delta D_t \sin(\theta_t) \end{bmatrix} \\ G_t &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ R_t &= V_t M_t V_t^T \\ V_t &= \frac{\partial g(\mu_{t-1}, u_t)}{\partial u_t} \\ V_t &= \begin{bmatrix} \cos(\theta_t) & -\Delta D_t \sin(\theta_t) \\ \sin(\theta_t) & \Delta D_t \cos(\theta_t) \end{bmatrix} \end{aligned}$$

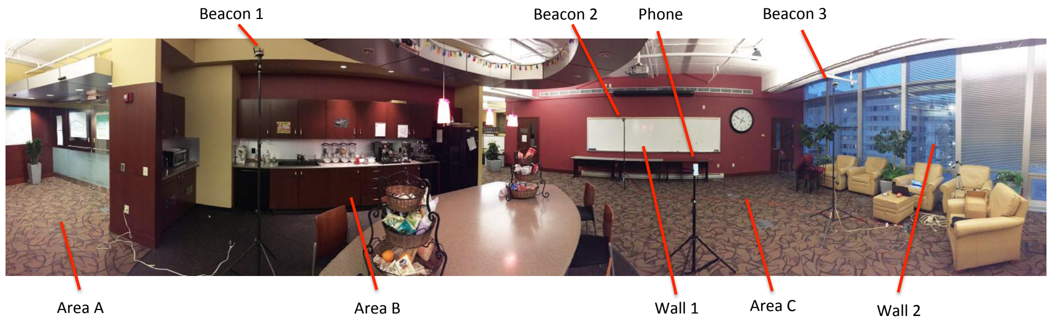


Figure 5.3: Panorama of automatically configured kitchen area using three beacons

5.3.2 Measurement model

Though the actual measurements from our system are the ToF range values from beacons, these can not be directly used with an EKF due to the linear approximation of the ToF equations. Instead, we first estimate the position using the range measurements, and use this estimate as our measurement. Our measurement model is given by:

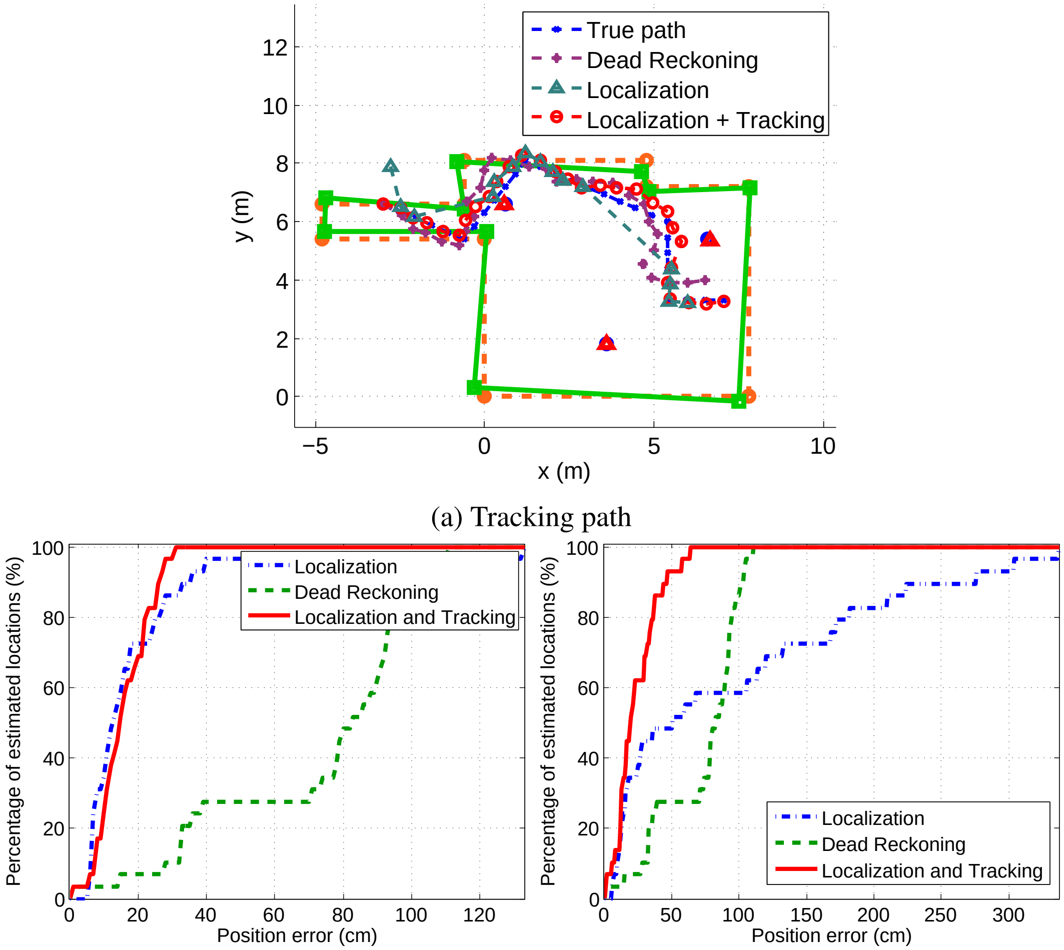
$$Z_t = \begin{bmatrix} x_t \\ y_t \end{bmatrix} + w_t$$
$$w_t \sim \mathcal{N}(0, Q_t)$$

where $Z_t = [\hat{x}_t, \hat{y}_t]^T$ is obtained by multilateration. We assume that the errors in the $\hat{x}_t$ and $\hat{y}_t$ are uncorrelated and for our ultrasonic system, assign $Q_t = \sigma_z I$ where $\sigma_z = 30cm$.

In case one or more beacons are blocked, or if the phone detects that one of the signals from the beacons is a NLOS signal, it does not update its measurement Z_t. In this case, we assign $Q_t = \sigma_n I$ where σ_n is a large number, such that the filtering effectively updates the estimate of the location based purely on tracking.

5.3.3 Evaluation

We evaluated our EKF-based tracking in half a dozen areas: a kitchen and lounge space, a lab and four office areas. The largest space in terms of area and number of corners was a lounge and kitchen space, as shown in Figure 5.3, with a total area of around 775 sq ft. and 10 corners. In each test, a user held an iPhone 5S and took approximately 30 steps in the area. We collected data from the compass and read step values. Range measurements were collected from ultrasonic beacons. Results from our largest scenario (the kitchen area) are presented in Figure 5.4. The *Localization* line refers to position estimated based on only the ranges from the beacons, the *Pedestrian Dead Reckoning* line refers to position estimates purely based on the IMU sensors and the motion model and the *Localization and Tracking* line refers to the output of the EKF explained above. Figure 5.4b shows that tracking does not improve the accuracy much as compared to using only localization, since the localization is much more accurate (error less than $30cm$ 90% of the time) than the estimates from the motion model. We then simulated situations when the user blocks one transmitter by removing some of the range measurements from a beacon in the



(a) Tracking path

(b) Localization accuracy without obstacles

(c) Localization accuracy with obstacles

Figure 5.4: Tracking performance in the kitchen with and without obstructions.

dataset. The *Localization* line in Figure 5.4a shows the localization estimates under this case. The location does not update when insufficient measurements are received. We observe that in such cases, the system benefits from tracking, as seen in Figure 5.4c, and the error is less than $50cm$ for 90% of the measurements.

5.4 Beacon mapping

In this section, we describe the beacon mapping algorithm and present the experimental results with UWB beacons.

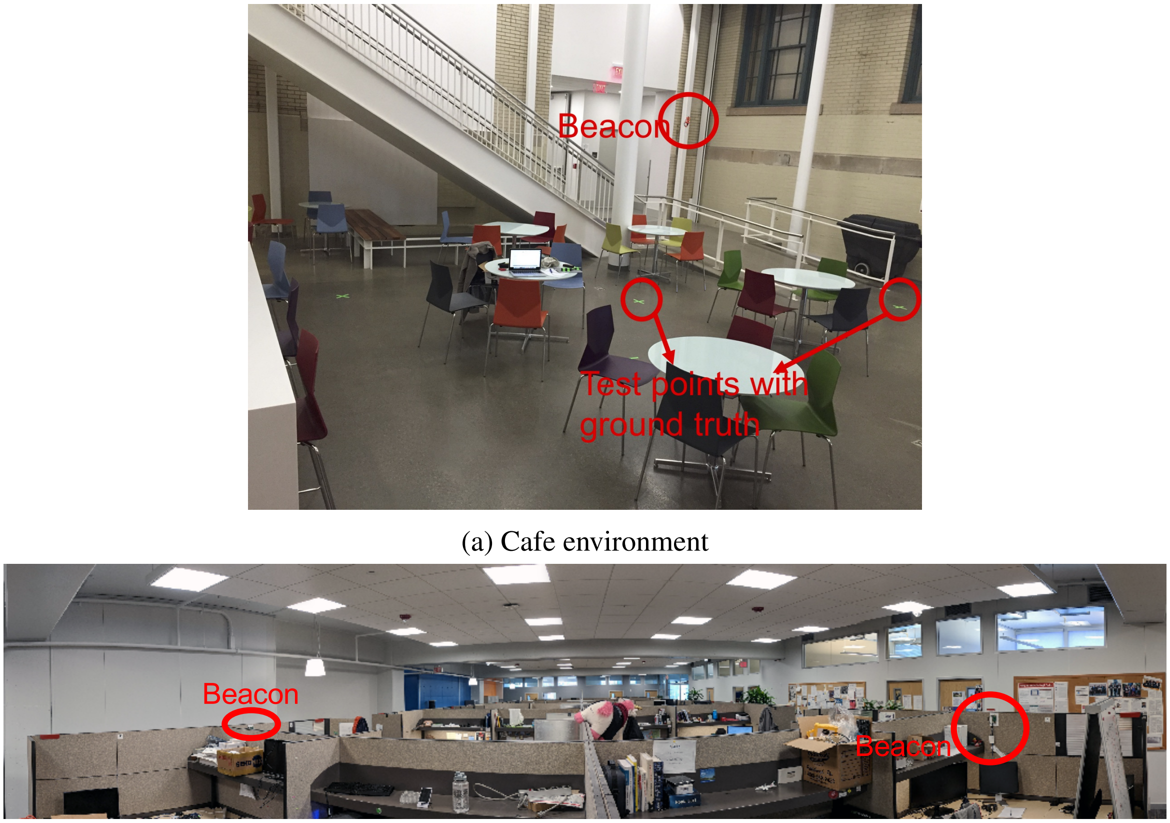
5.4.1 Approach

The beacon mapping process utilizes the Particle Filter-based sensor fusion approaches described in Section 5.2. In order to map the beacons with respect to each other, we draw upon existing work in RBPF-based SLAM where the beacon locations are conditionally independent given the location of the receiver. This approach has multiple advantages: (1) the complexity of the algorithm grows linearly with the number of beacons, (2) subsets of beacons can be updated simultaneously – for large indoor spaces, the beacons in various zones can be estimated independently by small traces of users walking through each zone and (3) it is a multiple-hypothesis approach which can accommodate non-linear motion models and non-Gaussian noise models. A particle filter is used to represent the belief of the mobile device where each beacon can be represented by any probabilistic distribution, such as a single Gaussian distribution, a Particle Filter or Sum-of-Gaussians. We adopted a purely probabilistic approach and represent each beacon's belief by a Particle Filter, as in [22]. While prior work has demonstrated the mapping performance with robotics systems, we implement and evaluate the mapping with a pedestrian-held mobile device by leveraging the accurate tracking provided by VIO. As an output of SLAM, we get the coordinates of the beacons with respect to each other and the VIO trace. The very first time the mapping is done, a preliminary beacon map is obtained, which can be used as a basis for adding and removing beacons over time. We use the same motion and measurement models described in Section 5.2 for fusion of beacon ranges and VIO. The main difference between the implementation of tracking and SLAM is that in SLAM, each particle that represents the mobile device has a belief of the beacons. The belief of beacons is initialized using the first few range measurements (this overcomes the problem of incorrect initialization when the first measurement is a NLOS range). Since low-range measurements are unlikely to be NLOS and can be trusted more than higher range measurements, when we get low range measurements (less than 3m in our implementation), we spawn a fraction of new particles. This helps in scenarios where the beacon belief is incorrect due to repeated NLOS signals.

5.4.2 Evaluation

To evaluate the beacon mapping process, we use Decawave UWB beacons. However, any of the various emerging TOF ranging technologies could work. We assume a ring-type of noise model, where the measured range is drawn from a uniform distribution of radius $+/-30cm$ about the true range. As part of the noise model, we allow a 50% likelihood that any particular range measurement is in error. This helps to account for erroneous range measurements resulting from noise or NLOS effects. For more complex noise models, an interested reader can refer to [3]. For each round of message exchange between the beacons and the tag, each beacon computes a range estimate between itself and the attached tag. For debugging and simplicity, the ranges are published by the beacons to a MQTT topic by each internet-connected beacon. The mobile device subscribes to this topic to collect the ranges and then runs the estimation algorithms locally. We evaluated our system in two environments: (1) a *Cafe* of size $145m^2$ ($1560ft^2$), shown in Figure 5.5a, which had five UWB beacons deployed, with most of the regions in the Cafe being in LOS and (2) an *Office* area of size $400m^2(4300ft^2)$, shown in Figure 5.5b, with 10 beacons deployed. All beacons were deployed inside the cubicles and all were in NLOS given the walkable areas. Test points and beacon locations were mapped accurately using a total station for ground truth. Four different users (of different heights and holding patterns) held a device in hand and walked at normal pace (4-5km/hr), clicking a button while crossing each test point.

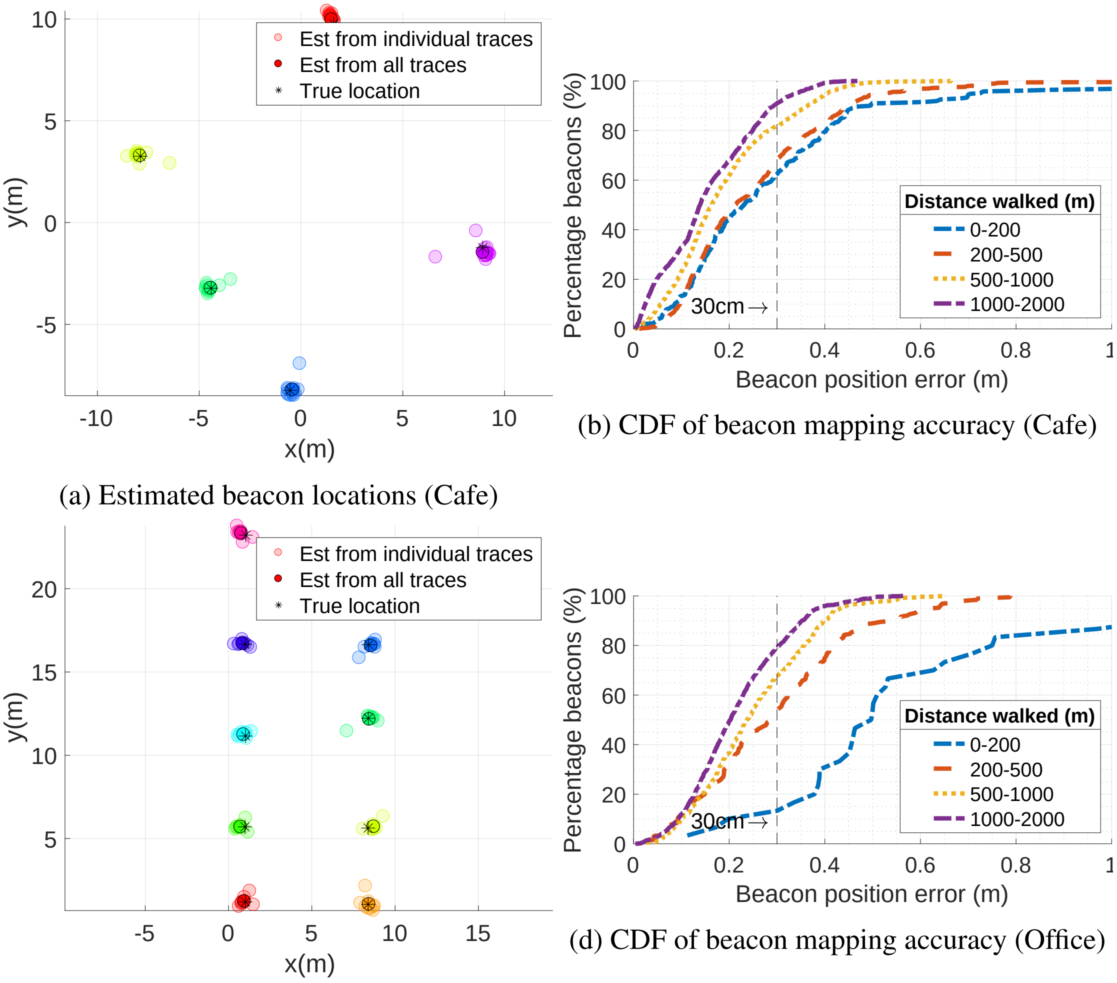
Figure 5.6 shows the result of the beacon mapping algorithm in the Cafe environment based on data from multiple traces. The individual traces vary in length from 30-250m in the Cafe. Fig-



(a) Cafe environment

(b) Office environment

Figure 5.5: Environments for testing beacon mapping and magnetic-field based orientation acquisition.



(a) Estimated beacon locations (Cafe)

(b) CDF of beacon mapping accuracy (Cafe)

(c) Estimated beacon locations (Office)

(d) CDF of beacon mapping accuracy (Office)

Figure 5.6: Beacon mapping performance

ure 5.6a shows the result of beacon mapping from the individual traces, and the estimate from combining all the traces. Next, we show the effect of distance walked by the user on the beacon mapping accuracy. We take combinations of the individual traces, and for each combination, estimate the total length of the traces and the mean beacon locations across traces. Figure 5.6b shows the beacon mapping performance for varying lengths of traces. Our algorithm achieved 90% error of $0.45m$ and $0.30m$ with $200m$ and $1800m$ of walking in the LOS Cafe environment, respectively. We see that longer trace lengths give better mapping performance, but the improvement is not significant after the user walks beyond 500m, which takes about seven minutes at normal walking pace. In the Office environment, where all the beacons were in NLOS from the walking paths, we were able to map with 90% error of $0.47m$ and $0.33m$ with $400m$ and $2400m$ of walking. We expect that in LOS conditions, where we expect a finite variance but no bias in the range measurements and hence the location estimates, the beacon locations would eventually converge to the true locations with sufficient mapping data. The initial beacon map can be built by an installer and subsequently improved by users in a crowdsourced manner.

5.5 Floor plans mapping

In this section, we present an approach for mapping the floor plans using a mobile device and range data from beacons. We map floor plans by using beacons to localize a mobile device, which a user places at key landmarks such as corners. The corners define the floor plan, and we can either interpolate between the corners to create the floor plan or track the device continuously to trace out the floor plan. This process can be performed by a non-expert user in a few minutes for a single area and has been adopted by companies like Estimote for their ultra-wideband mapping.

5.5.1 Procedure

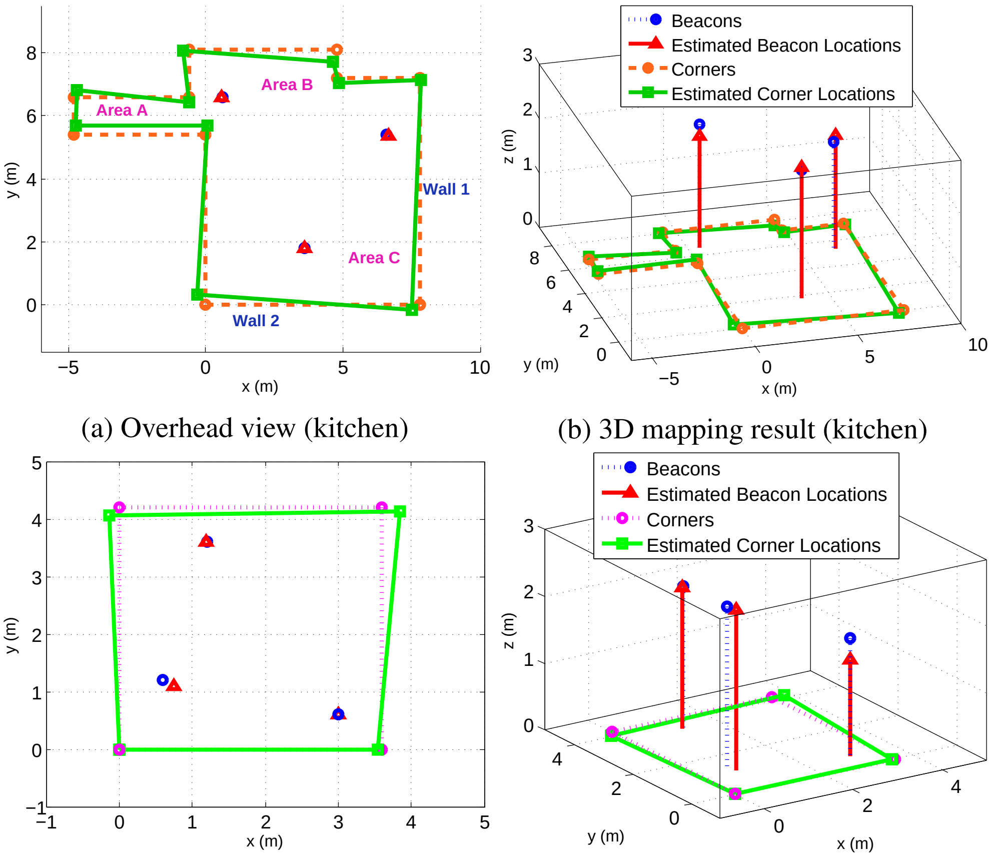
The process for mapping three beacons with inter-beacon ranging in a single area is given below. The approach can be extended to more beacons in a single area and, conceptually, also multiple areas. To extend the process to more beacons, we have to first map the beacons with respect to each other. Our process below includes the mapping of the three beacons in a single coordinate frame. We developed a mobile app that guides the user through these steps:

1. Deploy the three beacons such that they provide good coverage of the area and are in LOS of each other.
2. Identify three points on the floor such that all three beacons are visible from each point. Place the phone at each location, and select the *ground reference point* option.
3. Walk around the room and go to each corner and select the *corner reference points*. You can either choose to interpolate between corners, i.e., generate the floor plan using line segments between the corner points, or use device mobility and device tracking to create the floor plan regions between the corners. In our implementation, we opted for line segments between corner points.
4. Specify an origin and the orientation of the $x - y$ coordinate space. One way to do this is to select one of the corners as the origin and an adjacent corner to be on the x- or y-axis.

5.5.2 Algorithm

The basic principle of the 3D mapping process is that we make use of the following types of information to uniquely solve for the beacon positions and create the floor plan: (a) inter-beacon ranging, (b) ranging between beacons and a mobile device, (c) inertial tracking (optional), (d) estimation of $z - plane$ using the three ground measurement points and (e) user specified $x - y$ plane origin and orientation. The mapping algorithm for a scenario with three beacons is as follows:

1. Given inter-node ranges r_{12}, r_{23} and r_{13} between the three beacons B_1, B_2 and B_3, define a 3D coordinate system $\mathbb{R}_a^3$ such that the three beacons are on the $z = 0$ plane, B_1 is the origin $[0, 0, 0]$, and B_2 is along the x-axis $[r_{12}, 0, 0]$. Coordinates of B_3 can be obtained as $[r_{13}\cos(\alpha), r_{13}\sin(\alpha), 0]$, where $\alpha = \arccos(\frac{r_{12}^2+r_{13}^2-r_{23}^2}{2r_{12}r_{13}})$.
2. Estimate the coordinates of the three *ground reference points* with respect to the beacons in $\mathbb{R}_a^3$.



(a) Overhead view (kitchen)

(b) 3D mapping result (kitchen)

(c) Overhead view (lab)

(d) 3D mapping result (lab)

Figure 5.7: Qualitative results for beacon and floor plan mapping

3. Define a new coordinate system $\mathbb{R}_b^3$ such that the plane that contains the three ground points is the new $z = 0$ plane in $\mathbb{R}_b^3$.
4. The $x - y$ plane of $\mathbb{R}_b^3$ can be defined by its origin and one of the axes. The axes can be chosen arbitrarily, since we would re-assign the $x - y$ plane after generating the floor plan. In our implementation, we did the following: The projection of B_1 on the $x - y$ plane is assigned as the origin $(0, 0, 0)$ of $\mathbb{R}_b^3$. The projection of B_2 on this plane is assigned to lie on the y-axis of the new plane. The x-axis of $\mathbb{R}_b^3$ is found to be normal to the y and z axes.
5. Estimate the location of all the *corner reference points* in $\mathbb{R}_b^3$ using trilateration.
6. If line-segment interpolation between corner points is not desired, and the user wishes to create the floor plan lines in between the corner points, use a tracking method to create the segments between the corners.
7. The $x - y$ coordinates of the required 2D coordinate system are specified by the user during the calibration process. Either apply an affine transformation on $\mathbb{R}_b^3$ to get the final coordinate system, or for better accuracy, apply non-linear transformations to minimize error across all reference points if more than two reference points are available.

	Beacon Error (cm)				Corner Error (cm)	
Setup	**Avg.**	**x**	**y**	**z**	**Avg.**	**Max**
Kitchen	13.9	2.2	1.4	13.4	26.8	43.6
Lab	18.2	5.4	3.6	13.6	13.0	25.2
Office 1	17.5	4.6	3.5	15.0	10.7	13.9
Office 2	17.2	5.0	1.6	15.1	22.8	34.0
Office 3	15.5	2.3	1.7	11.1	18.9	40.9
Office 4	14.1	3.4	3.1	12.9	26.5	31.4
Overall	16.1	3.8	2.5	13.5	19.8	43.6

Table 5.1: Performance of beacon and floor plan mapping

5.5.3 Evaluation

Here, we present the results of the floor plan mapping process in the same environments described in Section 5.3.3. While mapping the floor plan using the proposed approach, we also map the beacons. We present the beacon mapping results as well. The generated map is shown in Figure 5.7. Note that this process requires all the corners to be in LOS of the three beacons. Some of the boundaries in Figure 5.7 were not physical walls but were either $1.5m$ tall partitions or were chosen to ensure all corners are in LOS. The results of the mapping process for the kitchen setup and averaged across all six experimental setups are shown in Table 5.1. Our system can determine three-dimensional beacon location with a Euclidean distance error of $16.1cm$ averaged over the three beacons, and can generate maps with room measurements with a two-dimensional Euclidean distance error of $19.8cm$ averaged over all the corners. We observe that while mapping the beacons, the overall error in the height is around $13.5cm$, while the error in the x or y coordinate is less than $4cm$. This is because the height of the beacons were within $1m$ of each other, whereas they were well separated in the $x - y$ plane. Hence the height is more sensitive to errors.

5.6 Summary

In this chapter, we presented approaches to fuse mobility data from phone sensors (inertial and camera) with beacon ranges for accurate tracking of a mobile device. We leverage this mobility for mapping beacons and mapping the floor plan information. With the ultrasonic platform, we mapped floor plans with an average two-dimensional Euclidean distance error of $19.8cm$ and beacons with an error of $16.1cm$ in the x-y plane. We demonstrated mapping of UWB beacons with range-only SLAM using VIO to a 90% accuracy of $0.45m$ and $0.30m$ with traces of length $200m$ and $1800m$ respectively, by a pedestrian walking in a LOS environment.

We believe that creating maps in an easy, cost-effective manner by crowdsourcing data from pedestrian-held phones will reduce the barrier to adoption of range-based beacon technologies.

Chapter 6

Orientation acquisition

In Chapter 4, we presented a robust location acquisition solver for the real world. For several applications, in addition to instant accurate location, we also require an instant accurate orientation estimate. However, range-based beacons do not provide any orientation information. In this chapter, we build on the mapping techniques presented in Chapter 5 and present a method to fuse beacon ranges, VIO and the magnetic field for rapid orientation acquisition on mobile devices [105].

Orientation acquisition outdoors can easily be accomplished using a magnetometer; since the earth's magnetic field generally points in a constant direction outdoors, it can be used as a global orientation reference. However, indoor magnetic fields tend to fluctuate wildly across space due to the metallic materials in the building structure and objects inside. We show that with accurate six-degrees-of-freedom tracking from VIO, we now have enough information to map the 3D magnetic field vector at a fine enough granularity to act as a calibration system for future users. Existing works that have constructed such maps of the magnetic field direction typically employ a robotic system with the sensor mounted at a known orientation, as well as accurate wheel odometry. These techniques are not practical for building-scale, due to the cost and effort involved. Leveraging accurate VIO, our approach is the first that is able to determine the full vector magnetic field map such that it can be used to calibrate a mobile phone or a compass mounted on a smart device held in any orientation. Our system maps the magnetic field by having users simply walk around wearing or holding the device normally. Subsequently, devices can instantly localize themselves using the beacons and instantly estimate their orientation using the on-board magnetic sensor and the previously obtained magnetic field map to drastically reduce vision-based search uncertainty. We demonstrate the accurate location and orientation obtained by our system with augmented reality applications that do not rely on visual point clouds for mapping.

In this chapter, we present a method for rapid pose acquisition that uses crowdsourced vectored (pose invariant) magnetic field maps. We experimentally show the performance of this approach across a variety of environments and over time. We present an end-to-end system implementation of accurate location and orientation acquisition using beacons and magnetic field, and apply this towards multi-user augmented reality applications.

6.1 Background and related work

In this section, we first describe the state-of-art approaches for relocalization using vision, then discuss approaches that use magnetic field for location and orientation estimation, and finally discuss prior work in creating spatially varying signal maps indoors.

6.1.1 Vision-based relocalization

Relocalization, also referred to as pose acquisition or registration, using vision based approaches is performed by matching features in the field of view with pre-mapped visual features. Several research studies have shown accurate relocalization using machine learning techniques [29, 60] and by fusing additional sensing modalities such as WiFi and magnetic field [28]. Most of these approaches have demonstrated accurate and efficient relocalization on datasets, but are computationally intensive for mobile devices. However, ARKit 2 by Apple and ARCore by Google have shown persistent AR by performing relocalization using pre-stored maps. Vision-based relocalization is promising, since it does not rely on external infrastructure. However, vision will always suffer in environments devoid of features, with changes in lighting, and where the scenery changes over time. In addition, vision-based relocalization in large areas requires searching through many candidate feature matches, which can become expensive if an initial location estimate is not provided. It is often the case that the user must walk around and view several areas of a scene before visual relocalization is able to take effect. We are optimistic that vision will continue to improve, but there are certain environments like office cubicles, hospitals or parts of airport terminals where even humans have trouble figuring out their location without exploring. Due to the limitations of a purely vision-based approach, we advocate combining visual approaches with range-based beacons.

6.1.2 Magnetic field sensing

Several prior works leverage the spatial variation and temporal stability of the magnetic field inside buildings for estimating location [25, 117, 123, 134, 136]. These approaches map the magnetic field magnitude as a function of location during the system setup and subsequently use it as a signature to estimate location. Since the magnetic field signature is not unique, it is integrated with other sensors or matched against a time series pattern as the user walks around [111, 117, 135, 136]. Our work differs from these approaches since we localize using beacons, and use the mapped magnetic field direction to estimate orientation.

Prior work has shown that orientation can be estimated using magnetic field by looking for opportunities when the change in orientation from the magnetometer is consistent with the gyroscope, indicating that the field is stable in that region [146]. The authors found on average two opportunities per minute when the field is stable. This requires the user to walk around and is not suitable for instant orientation acquisition. Robotic systems with sensors mounted in a known fixed orientation and accurate odometry use the magnitude and the direction of the magnetic field [61], but robots are neither affordable nor flexible enough for all applications. The closest to our work is [68], which presents a pedestrian-held system for magnetic field mapping. However, it relies on the phone being rigidly held and carried along a specific pre-determined route through the room. Our approach obtains the vectored magnetic field maps in any indoor space with a phone, where the user is free to walk through the room at any speed and any walking pattern.

6.1.3 Mapping spatially varying vectors indoors

Several algorithms for mapping spatially varying signals indoors overlap with the algorithms for creating indoor floor plan maps, as both aim to map features in the environment. The most common signatures mapped indoors are WiFi signal strength and magnetic field. One class of mapping systems uses robots or backpacks with sensors such as laser scanners, camera, inertial units and depth cameras to create maps of signal variation with locations [61, 71, 80, 98]. Our approach differs from these systems as we perform the mapping with phones held by pedestrians in any orientation. We are able to achieve this by fusing data from range-based beacons and VIO. VIO provides accurate pose tracking and the beacons provide global localization and correct for errors that accumulate over long distances with VIO.

6.2 Magnetic field mapping and orientation acquisition

Our goal is to acquire the orientation of the device in an external fixed frame defined by the beacons. We solve this estimation problem by fusing data from beacons, VIO and the magnetic field. In Section 5.2, we described the process of fusing data from beacons and VIO to estimate the six-degrees-of-freedom pose of the device in the beacon reference frame. We build on this estimation technique to create magnetic field maps (the approach also applies to any other signal map that varies with locations) when the user is mobile. While estimating the pose of the device, as described in Section 5.2, if the user starts at a location that has no prior magnetic map, the state estimation starts off with a belief that the orientation is uniformly distributed in $[0, 2\pi]$. Eventually, when the user has moved around and the particle filter converges, we can rely on the estimate of θ, or the rotation between the AR and beacon frames. We start building the magnetic field map at the locations that the user covers while walking by using the continuously estimated values of mobile device, θ, the magnetic field sensor data and the VIO orientation and position data. Subsequently, when the user is stationary and starts up the localization or augmented reality application, we use the beacon ranges to acquire the device location and use the magnetic field data and VIO orientation to acquire θ, the device orientation.

Our architecture for both mapping the magnetic field and subsequently using the mapped field to acquire orientation is shown in Figure 6.1. In addition to the two reference frames introduced earlier, namely F_{BEAC} and F_{AR}, we introduce a third reference frame, F_{DEV}, which is the frame of reference in which the inertial sensors are logged, i.e., it moves with the device. These three frames are shown in Figure 6.2.

We add to the notations in Section 5.2 to define:

- F_{BEAC}: Beacon frame of reference that is fixed over time across users.
- F_{AR}: AR frame of reference that is fixed for an AR session on a single mobile device.

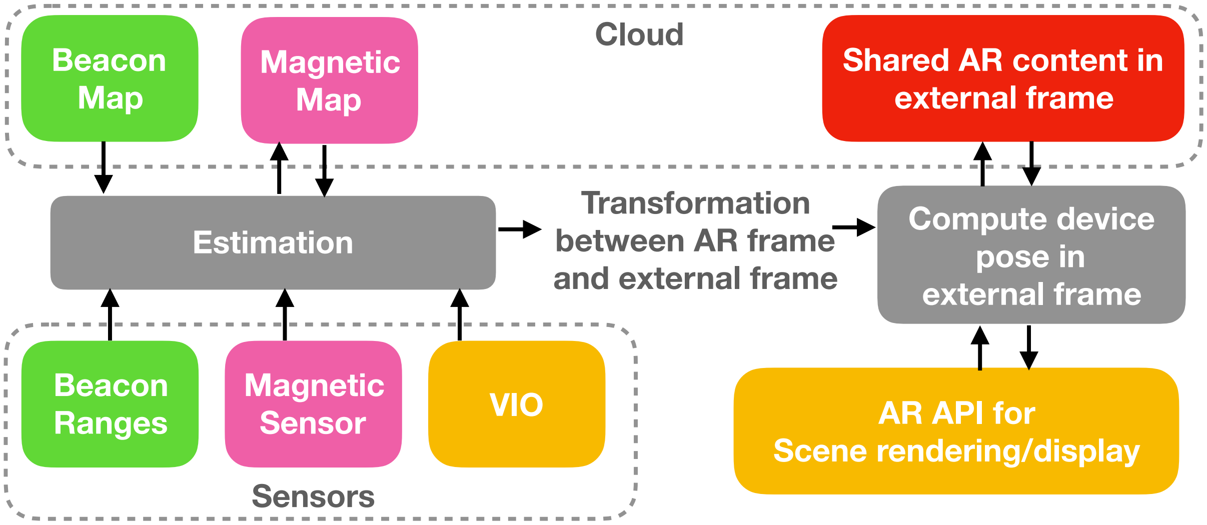

Cloud
Beacon Map
Magnetic Map
Shared AR content in external frame
Estimation
Transformation between AR frame and external frame
Compute device pose in external frame
Beacon Ranges
Magnetic Sensor
VIO
Sensors
AR API for Scene rendering/display

Figure 6.1: Architecture for fusion of beacon ranges, VIO and magnetic field

- F_{DEV}: Device frame of reference that moves with the device.
- M_{DEV}: Magnetic field vector in F_{DEV}.
- M_{AR}: Magnetic field vector in F_{AR}.
- M_{BEAC}: Magnetic field vector in F_{BEAC}.
- X_{AR} : 3D VIO position, i.e., x,y,z values of the device in F_{AR}.
- ϕ_{AR}: 3D VIO orientation, i.e., yaw, pitch, roll values of device in F_{AR}.
- X_{BEAC} : 3D position of the device in F_{BEAC}.

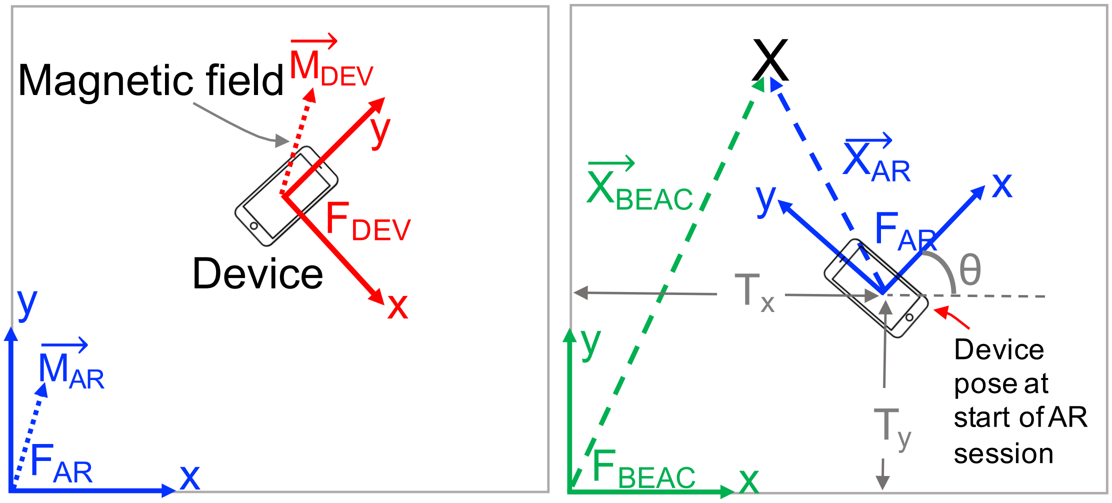
There is overlap between the steps to map the magnetic field and the steps to acquire the orientation. First, we describe the steps to map the magnetic field and then reuse the presented steps while describing the orientation acquisition process.

Step 1: Reading the magnetic field sensor

In this step, we simply read the 3D magnetic field M_{DEV} in the device frame of reference F_{DEV}. This frame is fixed with respect to the device as shown in Figure 6.2a.

Step 2: Converting the field from device frame F_{DEV} to AR frame F_{AR}

When the device moves in an AR session, the AR frame of reference F_{AR} remains constant but the device frame of reference F_{DEV} changes. In order to convert the magnetic field vector to an AR frame, we make use of the VIO orientation ϕ_{AR}. The VIO orientation gives us the roll, pitch and yaw of the device with respect to the AR frame. We transform the 3D magnetic field vector in device frame M_{DEV} by ϕ_{AR} to obtain the 3D magnetic field vector in AR frame M_{AR}. The result of this process is shown in Figure 6.3 with a view of the 3D magnetic field. When the device is completely stationary, we see that the magnetic field is mapped to a different vector from the device to AR frame. When the device is rotated about 360° in all orientations, the



(a) AR and device frames

(b) BEAC and AR frames

Figure 6.2: Frames of references: Beacon, AR and device frames

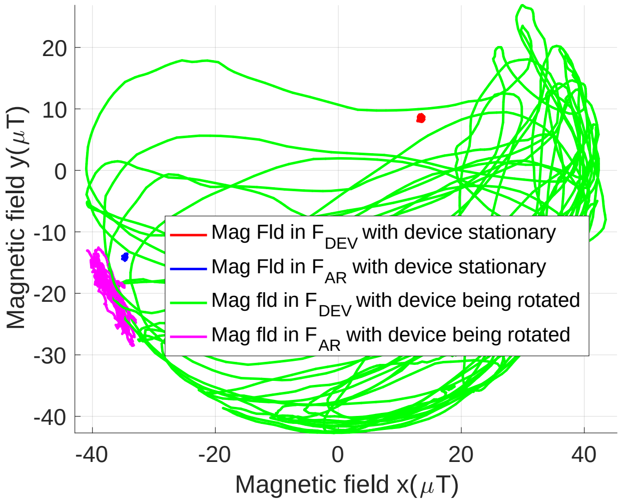
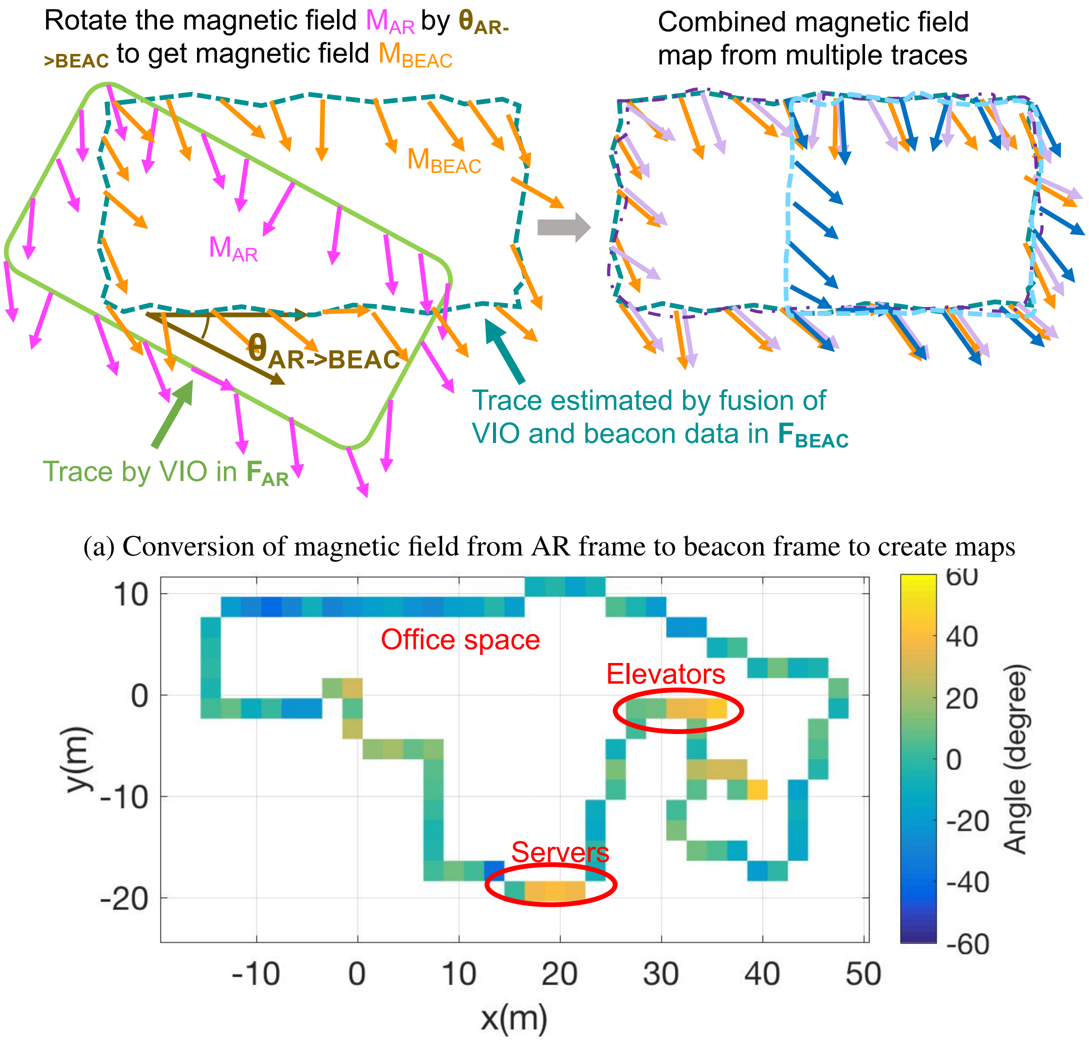


Figure 6.3: Conversion of the 3D magnetic field from device frame F_{DEV} to AR frame F_{AR}

magnetic field measured in the device frame rotates about a sphere but the magnetic field in the AR frame is fixed. In this manner we convert from M_{DEV} to M_{AR}.

Step 3: Converting the magnetic field from AR frame F_{AR} to beacon frame F_{BEAC}

We convert the magnetic field from the AR frame M_{AR} to the beacon frame M_{BEAC} using the estimated θ, which provides the rotation between the two frames of references about the z-axis. We project the magnetic field M_{AR} to 2D on the $x - y$ plane and then apply this rotation: M_{BEAC} = 2D projected $(M_{AR}) - \theta$. This process is illustrated in Figure 6.4a. The fusion estimates the rotation between the AR and beacon frame. This rotation is applied to the magnetic field to generate a vectored magnetic map. Several traces can be combined together over time as shown on the right in Figure 6.4a.



(a) Conversion of magnetic field from AR frame to beacon frame to create maps

(b) Resultant map of an office floor

Figure 6.4: Magnetic field mapping process

Step 4: Building the magnetic field map

Once we convert the magnetic field to the beacon frame in Step 3, we get the mapping between locations and the 3D magnetic field vector in the beacon frame. We store the map as a function of 2D locations. We evaluate the variation in height from 0.7 to $1.8m$ above ground (the height of standing and holding the device with arms outstretched down and holding it up) and found the z-variation to be within $10°$ at worst in a lab/cubicle space where there were monitors, metal cabinets, tables, chairs and devices. We elaborate on this in Section 6.3.1. Hence we store the map in 2D for it to be more tractable. We combine the field from multiple traces to build the map. The map at a location (x, y) is represented as the circular mean $M^{\mu}_{BEAC}(x, y)$ angle and circular standard deviation angle $M^{\sigma}_{BEAC}(x, y)$ of all the magnetic angles logged at location (x, y). Figure 6.4b shows an example of the real-world map of the magnetic field generated in an office floor of a building.

Orientation acquisition from the magnetic field map

To acquire the orientation upon startup, we use the previously generated magnetic field maps. The acquisition process differs from the mapping process only in the initialization of the state θ of the particles. First, the magnetic field is read in the device frame and converted to the AR frame using Steps 1-2 described above, to get M_{AR}. Then the magnetic field is read from the map at locations where the particles are initialized, i.e., we read $M^{\mu}_{BEAC}(x, y), M^{\sigma}_{BEAC}(x, y)$. While initializing particles, if no magnetic field information is available, we draw θ with uniform probability in $[0, 2\pi]$. Once the magnetic field information is available, we draw θ from a distribution based on the mean and variance of the mapped magnetic field at that location:

$$\theta \sim \mathcal{N}\big(M^{\mu}_{BEAC}(x, y) - M_{AR}, M^{\sigma}_{BEAC}(x, y)\big) \tag{6.1}$$

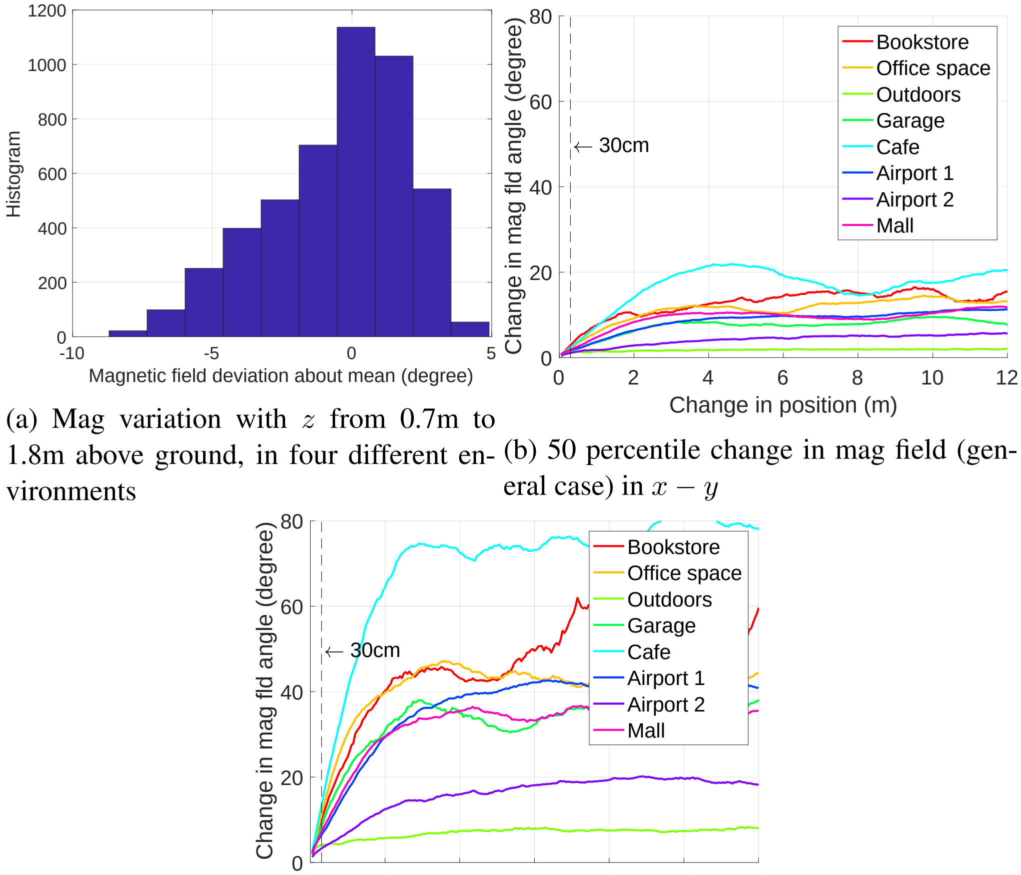
where (x, y) refers to the location of the particle at initialization. Each particle's θ dimension is initialized independently of other particles.

6.3 Evaluation

The experimental setup and environments for evaluating the magnetic field mapping and orientation acquisition are the same as described in Section 5.4.2. We used Decawave beacons and evaluated the system in the previously described Cafe and Office environments. First, we present the experimental evaluation of the feasibility of using the magnetic field for orientation acquisition. The magnetic field's spatial and temporal variation determines its feasibility for orientation acquisition.

6.3.1 Spatial variation

The spatial variation of the magnetic field along with the localization accuracy impacts the accuracy of the orientation acquisition. For instance, if the average localization error is $10cm$, and the magnetic field varies by around $50°$ for locations that are $10cm$ apart, then this spatial variation



(a) Mag variation with z from 0.7m to 1.8m above ground, in four different environments

(b) 50 percentile change in mag field (general case) in $x-y$

(c) 95 percentile change in mag field (worst case) in $x-y$

Figure 6.5: Magnetic field spatial variation

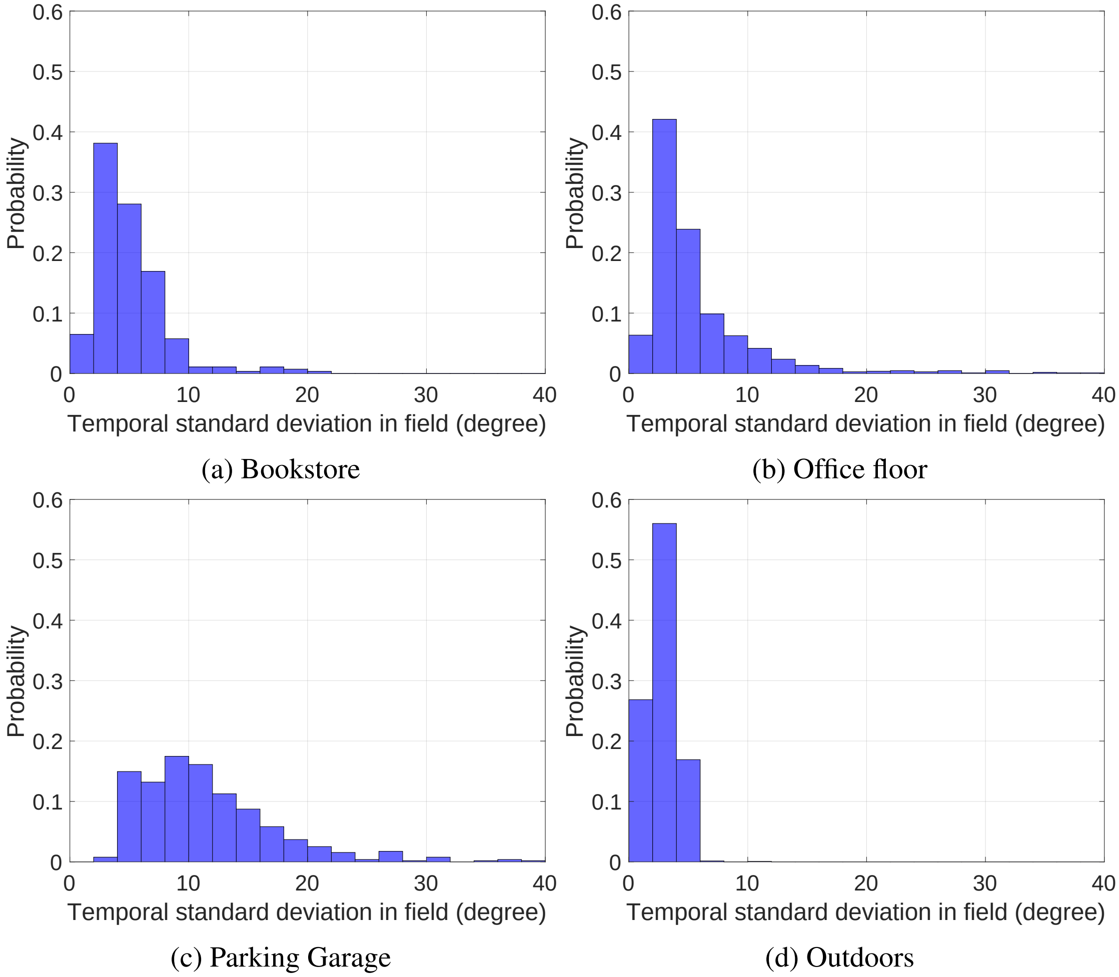
of the magnetic field along with the localization error could result in a $50°$ error in orientation acquisition. To test the spatial variation, we experimentally evaluated the magnetic field variation along the vertical (parallel to gravity vector) and horizontal axes. Figure 6.5a shows the variation along the z-axis for device height of $0.7m$ to $1.8m$ above ground ($1m$ about nominal standing height) across four environments (office, lobby, bookstore, cafe). Since the z-deviation is generally within $5°$, we store the magnetic field map in 2D, rather than 3D, to keep the map size tractable without losing significant accuracy. Next, we characterize the variation over horizontal distance in several environments, including outdoors, garage, mall, cafe and airports. We first compute the magnetic field angle difference between all pairs of locations that are a fixed distance apart. This captures the spatial variation for different distances. We then compute the CDF of the spatial variation for a fixed distance and extract the 50% spatial variation (representative of general case, shown in Figure 6.5b) and the 95% spatial variation (representative of worst case, shown in Figure 6.5c) for distances from $0 - 12m$. For instance, the *Cafe* line in Figure 6.5c shows that across all location pairs that are $1m$ apart, 95% have a field difference of less than $40°$. As one might expect, with a larger change in distance, there is a higher change in field across all locations. However, beyond a certain distance, these changes are random. The change in magnetic field for a distance corresponding to $30cm$ (the approximate positioning error from UWB localization system), is within 3° for 50% and varies from around $3 - 12°$ for the different environments for 95%. This reinforces the notion that the magnetic field can be highly variable indoors and that certain measurements generalize well across recorded paths while other areas need dense recordings.

6.3.2 Temporal variation

Since orientation acquisition relies on matching with a previously acquired map, the rate of change of field over time (temporal variation) impacts the orientation acquisition accuracy. Figure 6.6 shows the temporal variation over 10 weeks across four environments (book store, office floor, underground parking garage, outdoors). As expected, the outdoor environment has the least variation. The garage has the highest variation, due to cars (large metallic objects). In the two indoor environments, most regions have temporal standard deviations of less than $4°$. In some environments, like stores with moving metal shelves, the magnetic field will often change after a reconfiguration. Through crowdsourcing, our approach can continuously update and average magnetic field values from multiple users to adapt over time.

6.3.3 Localization accuracy

Before we acquire the orientation,we first acquire the device's location. Figure 6.7 shows the localization accuracy of the mobile device in both environments, Cafe with LOS and NLOS signals, and Office with only NLOS signals, by fusion of beacon ranges and VIO. Users walked continuously between test points, stopping for at least one second at each test point and pressing a button on the app. Since the users held the device in hand above the test point marked on ground, this introduced $10 - 15cm$ error between the device's true location and the test point that our experiment is limited by. In the Office environment and Cafe, we measure 80% error of $46cm$ and $27cm$ respectively.



(a) Bookstore

(b) Office floor

(c) Parking Garage

(d) Outdoors

Figure 6.6: Temporal variation over 10 weeks. The plots show the histogram of the circular standard deviation across time for all locations in the environment

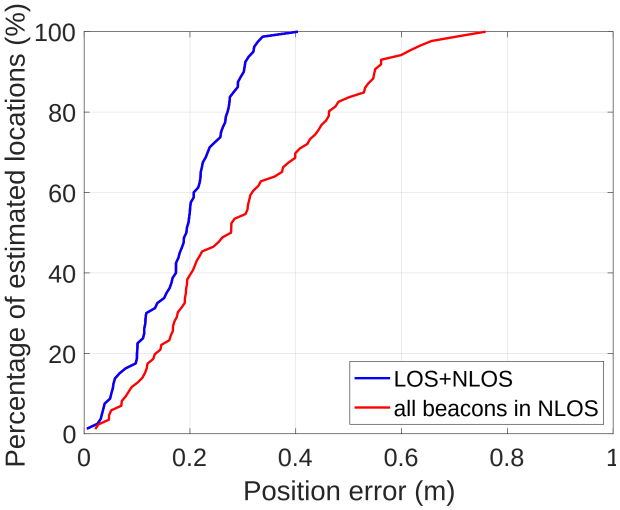
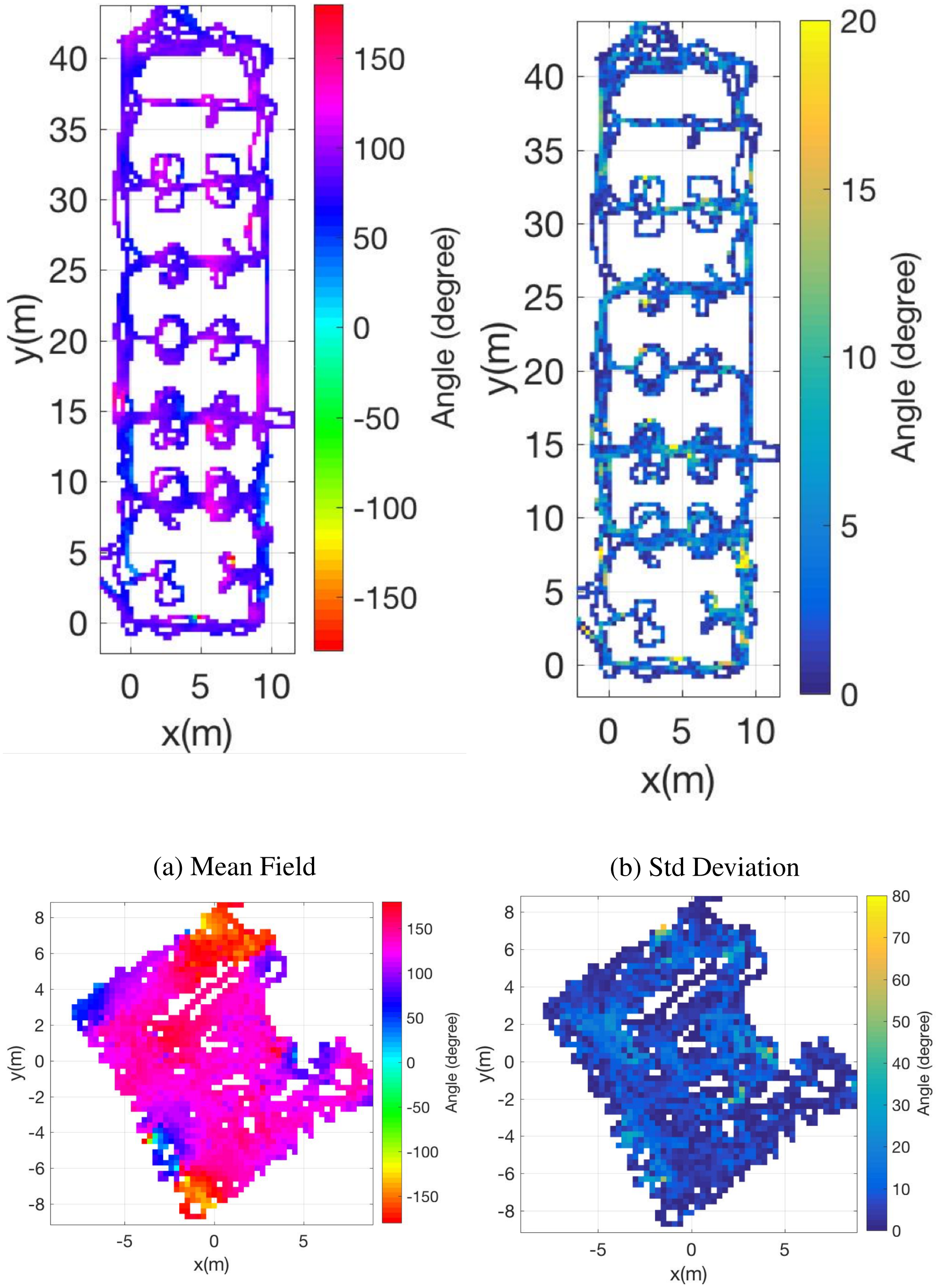


Figure 6.7: Location acquisition accuracy



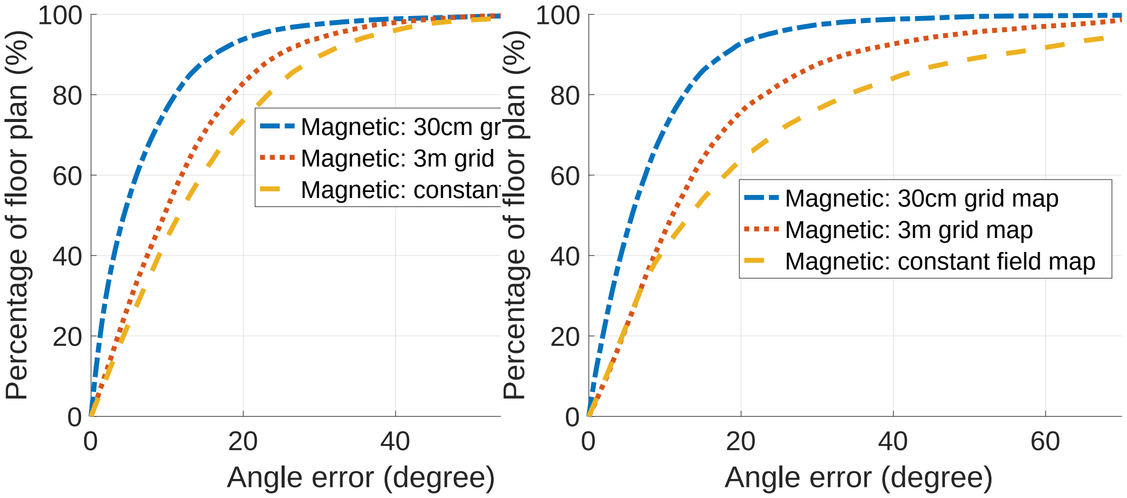
(a) Mean Field

(b) Std Deviation

(c) Mean Field

(d) Std Deviation

Figure 6.8: Magnetic field map captured in the two deployments. Office Space (a-b) and a Cafe (c-d)



(a) Orientation acquisition in Office space (b) Orientation acquisition in Cafe

Figure 6.9: Orientation acquisition accuracy

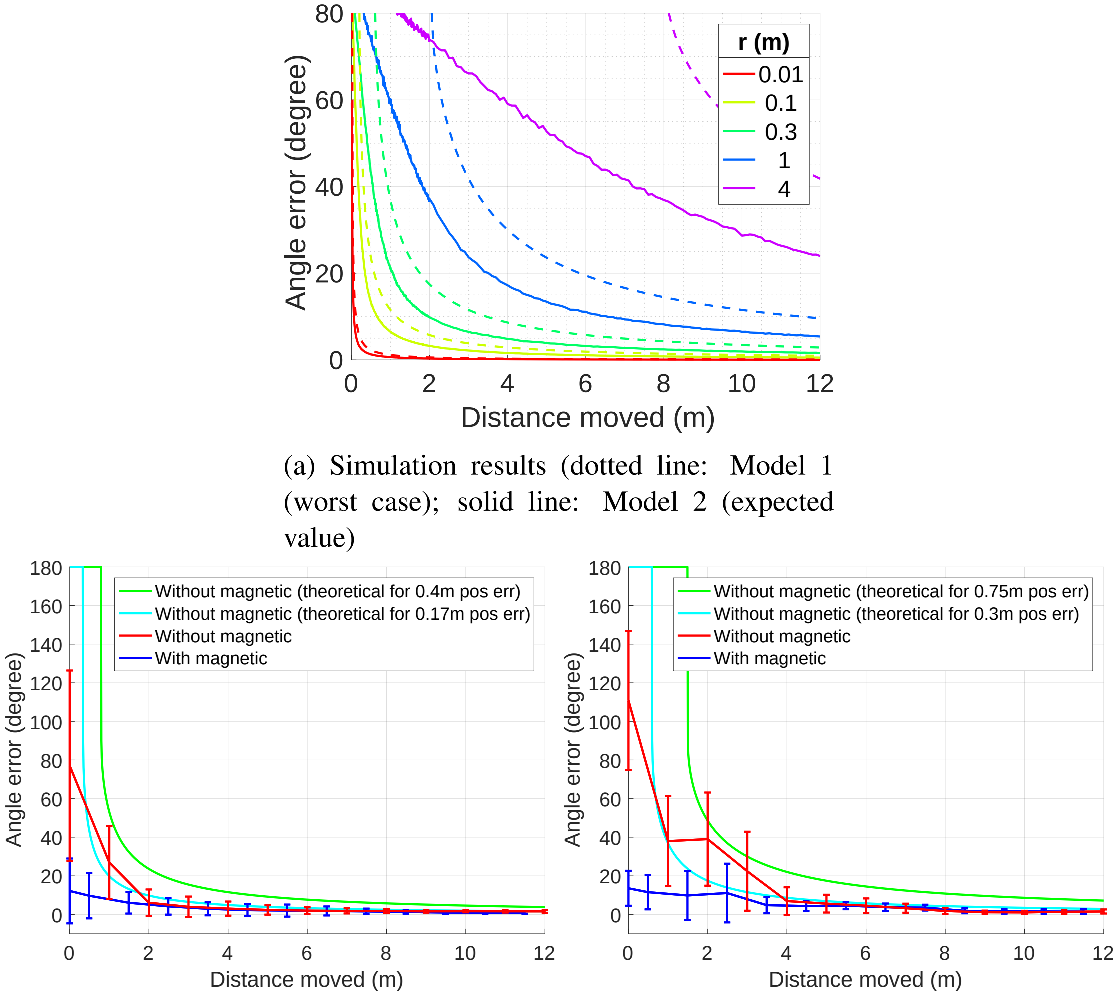
6.3.4 Magnetic field orientation accuracy

Figure 6.8 shows the magnetic field map in the two environments. The map is represented by the mean and standard deviation of the magnetic field. For the Office space, the standard deviation is $2°$ and $10°$ for 50% and 95% of the locations, respectively. For the Cafe, the standard deviation is $5.3°$ and $20.3°$ for 50% and 95% of the locations respectively for a grid size of $30cm$. As can be seen from Figure 6.5c, this Cafe has very high spatial variation compared to typical environments.

Figure 6.9 shows the accuracy of the magnetic-field based orientation for maps of varying spatial density. The *Magnetic: 30cm grid map* and *Magnetic: 3m grid map* show the angle error for two different grid sizes. The *Magnetic: constant field map* shows the performance if we assume the entire map has the same field, for which we used the mean of the field across the map. This is effectively equivalent to representing the map by a single sample. In the Office environment, we observe that the 80% point for a $30cm$ grid is $12.5°$ and with a constant field is $30°$. In the Cafe environment, we observe that the 80% point for a $30cm$ grid is $11.0°$ and with a constant field is $54°$. Across both environments, the 80% error is $11.7°$ for a $30cm$ grid. We also see that the $3m$ grid performance is closer to assuming a constant field than using a $30cm$ grid which highlights the utility of our approach in areas with high spatial variation. To reduce map storage, regions of the environment with high spatial variation should have a fine grid, while other regions can have a coarser grid. We also do not require complete coverage of a map, since a recorded trajectory can be extrapolated to support other nearby starting points with a confidence derived from the typical spatial variance captured in that area. We leave the compression of magnetic field maps to future work.

6.3.5 Distance moved to acquire orientation

In Section 6.3.4, we evaluated the accuracy of the orientation acquired upon startup. When the device moves, the orientation estimate is updated and over time (and distance) it converges to the correct orientation. In this section we answer the question, *"How much does the device need*



(a) Simulation results (dotted line: Model 1 (worst case); solid line: Model 2 (expected value)

(b) Orientation convergence over all traces in Cafe

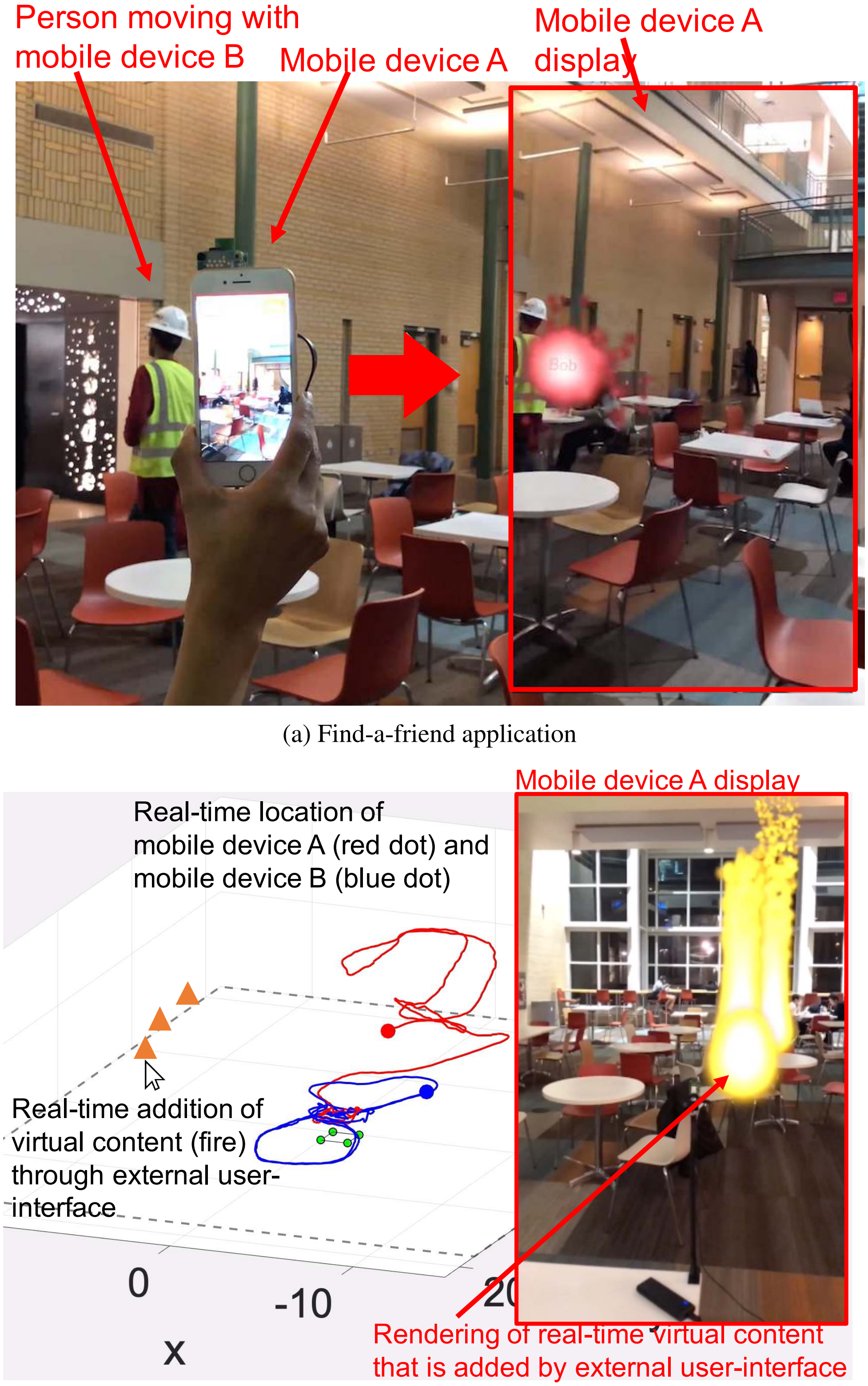
(c) Orientation convergence over all traces in Office

Figure 6.10: Orientation convergence with distance

to move before the orientation and location converge to the true values?" We answer this question in two parts: first, in simulation where we show the angle (orientation) convergence for two different localization error models with mobility but without magnetic field, and next, by experimentally showing the orientation convergence with and without magnetic field.

Theoretically, a device needs to move only between two unique locations in order to estimate the orientation using the approach with beacons and mobility. The accuracy of the angle estimation of the device is a function of the localization accuracy, which depends on several factors, including (1) the ranging error, (2) geometry of beacons, (3) the accuracy of the beacon locations and (4) frequency of beacon updates. The relationship between the location accuracy, ranging error and geometry of beacons for perfect beacon locations and synchronous beacon updates is captured by the Geometric Dilution of Precision (GDOP) metric [94, 121, 128]. We abstract away all of these parameters and assume that all locations have independent and identically distributed localization error. Consider a device that has moved between two locations that are at distance d apart. In the first model, we consider that the *Worst case* localization error at each location is given r. When the device moves by a distance d, the worse case angle error is determined by the tangent to the two circles of radius r separated by $d/2$ distance. This is given by $\sin^{-1}\frac{2r}{d}$. In the second model, the *Expected value* of the location error is drawn from a 2D zero mean Gaussian distribution with noise in each dimension being uncorrelated and standard deviation in each dimension as r. For this model, we generate the expected value of angle error through simulation by computing the mean angle error from randomly generated locations drawn from this distribution. The worst case and expected values are for different models, but for simplicity, we have shown the results of both in Figure 6.10a for different values of r. This gives a sense for how the performance would change with different localization technologies.

Figure 6.10b and Figure 6.10c show the performance of our system experimentally across several traces in both the test environments. We estimated the convergence time for each trace with error bars showing the distribution across all traces. Since the magnetic field-based orientation acquisition has a lower angle error at startup, it requires a lower distance from the mobility to achieve the same angle error, as compared to without the magnetic field. The implication of this is that a device instantly acquires an orientation, and when it moves a short distance it converges on the true orientation quicker than without using the magnetic field. Across both environments, we were able to get within 9° accuracy by moving the device $1m$. In theory, this could be achieved by waving the device around with a hand, in scenarios where a person holding the device is stationary at a location and requires a high orientation accuracy. We wish to point out here that the rate of convergence is also dependent on the parameters of the particle filter implementation, such as when resampling is performed, how frequently and when new particles are spawned and how much time/how many range measurements are required to initialize the filter upon startup. Nonetheless, using the magnetic field results in a faster convergence than without using the magnetic field.


Person moving with mobile device B
Mobile device A
Mobile device A display
Bob
Mobile device A display
Real-time location of mobile device A (red dot) and mobile device B (blue dot)
Real-time addition of virtual content (fire) through external user-interface
0
-10
20
x
Rendering of real-time virtual content that is added by external user-interface

(a) Find-a-friend application

(b) Real-time location-based commands in AR

Figure 6.11: Augmented reality demonstration application

6.3.6 Demonstration application

We built the end-to-end system and demonstrate it with an AR applications. As previously described in Section 1.5.1 of Chapter 1, we built mutli-user persistent AR applications with ultrasonic beacons. We also described the infrastructure-free firefighter localization in Section 1.5.2. We combine these to demonstrate a firefighter localization prototype system with augmented reality built with UWB beacons. Though we have demonstrated with mobile augmented reality, we envision that this approach could be incorporated into augmented reality displays worn by firefighters in future. A snapshot of the prototype system is shown in Figure 6.11. Our setup has four beacons deployed at fixed locations. Two persons representative of firefighters walk around with mobile devices attached to UWB tags. The devices are localized with respect to the beacons, and the real-time location pose information is available on an external console with a user interface. Figure 6.11a shows an application where both mobile devices are able to share persistent AR content that corresponds to the other device's location. We see that mobile device A is able to see the mobile device B in real-time in AR (with a red blob located around the mobile device B). This is representative of firefighters being able to see other virtually, through the building walls and smoke. Figure 6.11b shows the location of both devices made available on an external console. We see that a user is able to click at locations on the user interface (shown as orange triangles) and the mobile device is able to see new virtual content (fire) pop up in real-time at the user-defined locations. This is representative of a scenario where a fire hazard information is conveyed in real-time from a safety chief located outside the building, to firefighters' augmented reality headset displays. This prototype demonstrates the sharing of persistent AR content among the two mobile devices and the external system.

6.4 Summary

In this chapter, we presented a methodology to fuse beacon ranges, VIO and magnetic field to track the full pose of a mobile device, allowing us to record the full magnetic field vector and not just magnitude. This enables us to acquire the device orientation indoors in addition to the location. We show that with UWB ranging, we can acquire location with $80th$ percentile 3D accuracy of $27cm$ in LOS and $46cm$ in NLOS, and our magnetic field mapping approach can instantly estimate orientation with $80th$ percentile accuracy of $11.7°$.

Chapter 7

Conclusion

This dissertation presented a systematic methodology to integrate various sources of information to overcome the challenges faced by range-based localization systems, while trying achieve high performance at low cost. We implemented our approaches with two ranging platforms: an ultrasonic ranging platform and an ultra-wideband ranging platform.

This dissertation makes the following contributions:

1. **Beacon placement:** We present a systematic approach to integrate the floor plan geometry and beacon coverage models to reduce the number of beacons while maintaining localization coverage. We quantify the quality of beacon placements and use it to design beacon placement algorithms in an open source toolchain available to system designers. Our approach reduces the number of beacons on average by 33%.

2. **Mapping:** We present a crowdsourced pedestrian-aided mapping process that simply requires users to walk around with phones that can be held in any orientation. With the ultrasonic platform, we mapped floor plans with a two-dimensional Euclidean distance error of $19.8cm$ and beacons with an error of $16.1cm$ in the x-y plane. We demonstrated mapping of UWB beacons with range-only SLAM using VIO to a 90% accuracy of $0.45m$ and $0.30m$ with traces of length $200m$ and $1800m$ respectively, by a pedestrian walking in a LOS environment.
3. **Location acquisition:** We present a location solver that integrates the floor plan geometry and beacon coverage models to localize robustly in under-defined scenarios. Our solver shifts the 80% accuracy point from $48m$ to $1m$ as compared to solvers that do not use the floor plan information. We are able to detect and remove NLOS signals with 91.5% accuracy.
4. **Orientation acquisition:** We present a method to acquire device orientation in addition to location, by fusing beacon ranges, VIO and the magnetic field sensor data. We show that with UWB ranging, we can acquire location with $80th$ percentile 3D accuracy of $27cm$ in LOS and $46cm$ in NLOS, and our magnetic field mapping approach can instantly estimate orientation with $80th$ percentile accuracy of 11.7°.

7.1 Future directions

Looking ahead, several challenges remain to be solved for range-based localization systems. This dissertation has assumed that devices receive range measurements from beacons with known identity and location. In the future, we envision that with the emerging standards for ranging technologies, WiFi access points, mobile devices, smart home appliances and internet-of-things devices would be capable of ranging to each other. For these systems to range to a large number of devices in a single collision domain, we require time synchronization between the devices. For RF ranging technologies, devices require time synchronization on the order of nanoseconds. Further, multiple access and device discovery is non-trivial with heterogeneous ranging technologies in an ecosystem where devices' identities are not known.

In this dissertation, we have performed sensor fusion at an algorithmic level. We gather the available sensor data and generate map and location estimates. In the future, it may well be possible to integrate the sensor data at a lower level where we only sense when required. For instance, we can take a cross-layer approach where the quality of localization service requested by the application determines when to sense and what data to sense. The goal would be to minimize on-device resources (e.g., energy, computation) while meeting the requested quality of location service.

In order for indoor localization to be successful at scale, we need to design human-in-the-loop mapping processes where users get feedback on the quality of maps and are guided in order to reduce the uncertainty in the mapping process. A possible direction is to use augmented reality to provide the user visual feedback on the quality of the maps by overlaying the estimated map on the real world. We also require tools that can continuously detect if beacons have failed or moved and give system installers and users feedback about the performance in real-time.

In this dissertation, we have assumed a non-adversarial setting. We have assumed that the infrastructure and devices trust each other. In round-trip-time-of-flight ranging systems such as ultra-wideband, we have also assumed that the devices and infrastructure are aware of each other's identities. However, attacks on the ranging systems can reduce and enlarge range measurements. Here, the challenge is that the attacks may be indistinguishable from noise. A possible approach to overcome this could be to fuse different sensors' information to be more robust to attacks.

Finally, what we require is a unifying framework for indoor localization that accommodates current and future technologies, where various maps, real-time sensor data and models are integrated to provide various quality of services for localization based on the available sensing, power and compute resources. Devices in the environment should be able to subscribe to secure location services in a manner that preserves their privacy. This dissertation has taken a step towards this by contributing to a framework for range-based localization systems.

Bibliography

[1] https://developers.google.com/ar/reference/java/com/google/ar/core/Config.UpdateMode (viewed 3/10/2018). 70

[2] https://developer.apple.com/documentation/scenekit/scnview/1621205-preferredframespersecond (viewed 3/10/2018). 70

[3] Decawave Application Notes: https://www.decawave.com/application-notes (viewed 12/23/2017). 76

[4] Amazon Go: https://www.amazon.com/b?node=16008589011 (viewed 2/13/2019). 3

[5] https://developer.apple.com/documentation/arkit/arconfiguration.worldalignment (viewed on 12/28/2017). 68

[6] https://www.decawave.com/products/dwm1001-module (viewed 3/10/2018). 18

[7] Decawave error sources: https://www.decawave.com. 3

[8] https://www.3db-access.com/Product.3.html. 4

[9] https://www.wi-fi.org/news-events/newsroom/wi-fi-certified-location-brings-wi-fi-indoor-positioning-capabilities (viewed 10 Oct 2018). 4

[10] 802.15.4z: http://www.ieee802.org/15/pub/TG4z.html (viewed 2/13/2019). 4

[11] M. Addlesee, R. Curwen, S. Hodges, J. Newman, P. Steggles, A. Ward, and A. Hopper. Implementing a sentient computing system. *Computer*, 34(8):50–56, 2001. 43

[12] F. Adib, Z. Kabelac, and D. Katabi. Multi-person localization via {RF} body reflections. In *12th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 15)*, pages 279–292, 2015. 3

[13] R. Allen, N. MacMillan, D. Marinakis, R. I. Nishat, R. Rahman, and S. Whitesides. The range beacon placement problem for robot navigation. In *Computer and Robot Vision (CRV), 2014 Canadian Conference on*, pages 151–158. IEEE, 2014. 21

[14] J. Aspnes, T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. Anderson, and P. N. Belhumeur. A theory of network localization. *IEEE Transactions on Mobile Computing*, 5(12):1663–1678, 2006. 65

[15] A. Baggio and K. Langendoen. Monte carlo localization for mobile wireless sensor networks. *Ad hoc networks*, 6(5):718–733, 2008. 65

[16] P. Bahl and V. Padmanabhan. Radar: an in-building rf-based user location and tracking system. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and*

Communications Societies (INFOCOM '00), volume 2, pages 775 –784, 2000. doi: 10.1109/INFCOM.2000.832252. 2

[17] G. Balogh, A. Ledeczi, M. Maróti, and G. Simon. Time of arrival data fusion for source localization. In *Proceedings of The WICON Workshop on Information Fusion and Dissemination in Wireless Sensor Networks (SensorFusion 2005)*, 2005. 42

[18] A. N. Bishop, B. Fidan, B. D. Anderson, K. Doğançay, and P. N. Pathirana. Optimality analysis of sensor-target localization geometries. *Automatica*, 46(3):479–492, 2010. 21

[19] P. Biswas and Y. Ye. Semidefinite programming for ad hoc wireless sensor network localization. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 46–54. ACM, 2004. 65

[20] I. Bjorling-Sachs and D. L. Souvaine. An efficient algorithm for guard placement in polygons with holes. *Discrete & Computational Geometry*, 13(1):77–109, 1995. 21

[21] J.-L. Blanco, J.-A. Fernández-Madrigal, and J. González. Efficient probabilistic range-only slam. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1017–1022. IEEE, 2008. 66

[22] J.-L. Blanco, J. González, and J.-A. Fernández-Madrigal. A pure probabilistic approach to range-only slam. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1436–1441. IEEE, 2008. 66, 76

[23] G. Blewitt. Basics of the gps technique: observation equations. *Geodetic applications of GPS*, pages 10–54, 1997. 41

[24] F. Caballero, L. Merino, and A. Ollero. A general gaussian-mixture approach for range-only mapping using multiple hypotheses. In *2010 IEEE International Conference on Robotics and Automation*, pages 4404–4409. IEEE, 2010. 66

[25] D. Caruso, A. Eudes, M. Sanfourche, D. Vissière, and G. Le Besnerais. A robust indoor/outdoor navigation filter fusing data from vision and magneto-inertial measurement unit. *Sensors*, 17(12):2795, 2017. 83

[26] P.-C. Chen. A non-line-of-sight error mitigation algorithm in location estimation. In *Wireless Communications and Networking Conference, 1999. WCNC. 1999 IEEE*, volume 1, pages 316–320. IEEE, 1999. 42

[27] V. Chvatal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory, Series B*, 18(1):39–41, 1975. 21

[28] R. Clark, S. Wang, H. Wen, N. Trigoni, and A. Markham. Increasing the efficiency of 6-dof visual localization using multi-modal sensory data. In *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, pages 973–980. IEEE, 2016. 83

[29] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, 2017. 83

[30] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2009. 21

[31] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6):1052–1067, 2007. 3

[32] A. A. Diakité and S. Zlatanova. First experiments with the tango tablet for indoor scanning. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3:67, 2016. 2

[33] J. Djugash and S. Singh. A robust method of localization and mapping using only range. In *Experimental Robotics*, pages 341–351. Springer, 2009. 66

[34] J. Djugash, S. Singh, G. Kantor, and W. Zhang. Range-only slam for robots operating cooperatively with sensor networks. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2078–2084. IEEE, 2006. 65

[35] A. Doucet, N. De Freitas, K. Murphy, and S. Russell. Rao-blackwellised particle filtering for dynamic bayesian networks. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 176–183. Morgan Kaufmann Publishers Inc., 2000. 66

[36] S. Eidenbenz, C. Stamm, and P. Widmayer. Inapproximability results for guarding polygons and terrains. *Algorithmica*, 31(1):79–113, 2001. 21

[37] A. Ens, F. Höflinger, J. Wendeberg, J. Hoppe, R. Zhang, A. Bannoura, L. M. Reindl, and C. Schindelhauer. Acoustic self-calibrating system for indoor smart phone tracking. *International Journal of Navigation and Observation*, 2015, 2015. 43, 58

[38] T. Eren, O. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. Anderson, and P. N. Belhumeur. Rigidity, computation, and randomization in network localization. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2673–2684. IEEE, 2004. 65

[39] F. R. Fabresse, F. Caballero, I. Maza, and A. Ollero. Undelayed 3d ro-slam based on gaussian-mixture and reduced spherical parametrization. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1555–1561. IEEE, 2013. 66

[40] B. T. Fang. Simple solutions for hyperbolic and related position fixes. *IEEE transactions on aerospace and electronic systems*, 26(5):748–753, 1990. 41

[41] W. H. Foy. Position-location solutions by taylor-series estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 12:187–194, Mar. 1976. doi: 10.1109/TAES.1976.308294. 41, 42

[42] M. Geyer and A. Daskalakis. Solving passive multilateration equations using bancroft's algorithm. In *17th DASC. AIAA/IEEE/SAE. Digital Avionics Systems Conference. Proceedings (Cat. No. 98CH36267)*, volume 2, pages F41–1. IEEE, 1998. 41

[43] S. K. Ghosh. Approximation algorithms for art gallery problems in polygons. *Discrete Applied Mathematics*, 158(6):718–722, 2010. 21

[44] D. K. Goldenberg, P. Bihler, M. Cao, J. Fang, B. Anderson, A. S. Morse, and Y. R. Yang. Localization in sparse networks using sweeps. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 110–121. ACM, 2006. 65

[45] F. Hammer, M. Pichler, H. Fenzl, A. Gebhard, and C. Hesch. An acoustic position estimation prototype system for underground mining safety. *Applied Acoustics*, 92:61–74, 2015. 42, 43

[46] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, 43(5):1318–1334, 2013. 3

[47] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *The International Journal of Robotics Research*, 31(5):647–663, 2012. 66

[48] J. Herrera, A. Hinkenjann, P. Ploger, and J. Maiero. Robust indoor localization using optimal fusion filter for sensors and map layout information. In *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, pages 1–8. IEEE, 2013. 44

[49] 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*, pages 147–158. ACM, 2014. 44

[50] F. Hoffmann, M. Kaufmann, and K. Kriegel. The art gallery theorem for polygons with holes. In *Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on*, pages 39–48. IEEE, 1991. 21

[51] L. Hu and D. Evans. Localization for mobile sensor networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 45–57. ACM, 2004. 65

[52] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal. Efficient, generalized indoor wifi graphslam. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1038–1043. IEEE, 2011. 65

[53] Y.-H. Jo, J.-Y. Lee, D.-H. Ha, and S.-H. Kang. Accuracy enhancement for uwb indoor positioning using ray tracing. In *Position, Location, and Navigation Symposium, 2006 IEEE/ION*, pages 565–568. IEEE, 2006. 44

[54] J. Jung and H. Myung. Indoor localization using particle filter and map-based nlos ranging model. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5185–5190. IEEE, 2011. 44

[55] G. Kantor and S. Singh. Preliminary results in range-only localization and mapping. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 1818–1823. IEEE, 2002. 65

[56] E. Kaplan and C. Hegarty. *Understanding GPS: principles and applications*. Artech house, 2005. 16, 17

[57] S. M. Kay. Fundamentals of statistical signal processing, volume i: estimation theory. 1993. 21, 30

[58] B. Kempke, P. Pannuto, and P. Dutta. Polypoint: Guiding indoor quadrotors with ultra-wideband localization. In *Proceedings of the 2nd International Workshop on Hot Topics in Wireless*, pages 16–20. ACM, 2015. 2

[59] B. Kempke, P. Pannuto, B. Campbell, and P. Dutta. Surepoint: Exploiting ultra wideband flooding and diversity to provide robust, scalable, high-fidelity indoor localization. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 137–149. ACM, 2016. 2

[60] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 2, 83

[61] H.-S. Kim, W. Seo, and K.-R. Baek. Indoor positioning system using magnetic field map navigation and an encoder system. *Sensors*, 17(3):651, 2017. 83, 84

[62] N. Kirchhof. Optimal placement of multiple sensors for localization applications. In *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*, pages 1–10. IEEE, 2013. 22

[63] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 447–458. ACM, 2014. 2

[64] D. Kurth, G. Kantor, and S. Singh. Experimental results in range-only localization with radio. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 974–979. IEEE, 2003. 65

[65] P. Lazik and A. Rowe. Indoor pseudo-ranging of mobile devices using ultrasonic chirps. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 99–112. ACM, 2012. 2, 14, 43

[66] P. Lazik, N. Rajagopal, O. Shih, B. Sinopoli, and A. Rowe. Alps: A bluetooth and ultrasound platform for mapping and localization. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*, pages 73–84. ACM, 2015. 4, 14, 26, 43, 72

[67] P. Lazik, N. Rajagopal, B. Sinopoli, and A. Rowe. Ultrasonic time synchronization and ranging on smartphones. In *21st IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 108–118. IEEE, 2015. 2, 14, 18

[68] E. Le Grand and S. Thrun. 3-axis magnetic field mapping and fusion for indoor localization. In *Multisensor Fusion and Integration for Intelligent Systems (MFI), 2012 IEEE Conference on*, pages 358–364. IEEE, 2012. 83

[69] D.-T. Lee and A. K. Lin. Computational complexity of art gallery problems. *Information Theory, IEEE Transactions on*, 32(2):276–282, 1986. 21

[70] T. Leune, T. Wehs, M. Janssen, C. Koch, and G. von Colln. Optimization of wireless locating in complex environments by placement of anchor nodes with evolutionary algorithms. In *Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on*, pages 1–6. IEEE, 2013. 22

[71] P. Levchev, M. N. Krishnan, C. Yu, J. Menke, and A. Zakhor. Simultaneous fingerprinting and mapping for multimodal image and wifi indoor positioning. In *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 442–450. IEEE, 2014. 84

[72] X. Li. An iterative nlos mitigation algorithm for location estimation in sensor networks. *Proceedings of the15 th IST Mobile and Wireless Communications Summit*, pages 1–5, 2006. 25, 42

[73] H. Liu, G. Zhang, and H. Bao. Robust keyframe-based monocular slam for augmented reality. In *Mixed and Augmented Reality (ISMAR), 2016 IEEE International Symposium on*, pages 1–10. IEEE, 2016. 2

[74] K. Liu, X. Liu, and X. Li. Guoguo: Enabling fine-grained indoor localization via smartphone. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 235–248. ACM, 2013. 42, 43, 58

[75] K. Liu, X. Liu, L. Xie, and X. Li. Towards accurate acoustic localization on a smartphone. In *INFOCOM, 2013 Proceedings IEEE*, pages 495–499. IEEE, 2013. 43

[76] C. X. Lu, Y. Li, P. Zhao, C. Chen, L. Xie, H. Wen, R. Tan, and N. Trigoni. Simultaneous localization and mapping with power network electromagnetic field. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 607–622. ACM, 2018. 2

[77] Z. Luo, Q. Zhang, Y. Ma, M. Singh, and F. Adib. 3d backscatter localization for fine-grained robotics. In *16th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 19)*, pages 765–782, 2019. 2

[78] D. E. Manolakis. Efficient solution and performance analysis of 3-d position estimation by trilateration. *IEEE Transactions on Aerospace and Electronic Systems*, 32(4):1239–1248, Oct 1996. ISSN 0018-9251. doi: 10.1109/7.543845. 42

[79] I. A. Mantilla-Gaviria, M. Leonardi, G. Galati, J. V. Balbastre-Tejedor, and E. d. L. R. Davó. Efficient location strategy for airport surveillance using mode-s multilateration systems. *International Journal of Microwave and Wireless Technologies*, 4(2):209–216, 2012. 41

[80] P. Mirowski, R. Palaniappan, and T. K. Ho. Depth camera slam on a low-cost wifi mapping robot. In *2012 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pages 1–6. IEEE, 2012. 84

[81] P. Misra and P. Enge. *Global Positioning System: Signals, Measurements and Performance Second Edition.* Lincoln, MA: Ganga-Jamuna Press, 2006. 19, 30

[82] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. 2002. 66

[83] D. Moore, J. Leonard, D. Rus, and S. Teller. Robust distributed network localization with noisy range measurements. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 50–61. ACM, 2004. 65

[84] R. Nandakumar, V. Iyer, and S. Gollakota. 3d localization for sub-centimeter sized devices. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 108–119. ACM, 2018. 2

[85] W. Navidi, W. S. Murphy, and W. Hereman. Statistical methods in surveying by trilateration. *Computational Statistics and Data Analysis*, 27(2):209 – 227, 1998. ISSN

0167-9473. 42

[86] P. Newman and J. Leonard. Pure range-only sub-sea slam. In *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, volume 2, pages 1921–1926. Ieee, 2003. 65

[87] G. Oberholzer, P. Sommer, and R. Wattenhofer. Spiderbat: Augmenting wireless sensor networks with distance and angle information. In *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, pages 211–222. IEEE, 2011. 43

[88] B. Okorn, X. Xiong, B. Akinci, and D. Huber. Toward automated modeling of floor plans. In *Proceedings of the symposium on 3D data processing, visualization and transmission*, volume 2, 2010. 66

[89] E. Olson, J. J. Leonard, and S. Teller. Robust range-only beacon localization. *Oceanic Engineering, IEEE Journal of*, 31(4):949–958, 2006. 65

[90] J. O'rourke. *Art gallery theorems and algorithms*, volume 57. Oxford University Press Oxford, 1987. 21

[91] P. Pannuto, B. Kempke, L.-X. Chuo, D. Blaauw, and P. Dutta. Harmonium: Ultra wideband pulse generation with bandstitched recovery for fast, accurate, and robust indoor localization. *ACM Transactions on Sensor Networks (TOSN)*, 14(2):11, 2018. 2

[92] P. Pannuto, B. Kempke, and P. Dutta. Slocalization: sub-μw ultra wideband backscatter localization. In *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 242–253. IEEE Press, 2018. 2

[93] N. Patwari, A. O. Hero III, M. Perkins, N. S. Correal, and R. J. O'dea. Relative location estimation in wireless sensor networks. *Signal Processing, IEEE Transactions on*, 51(8): 2137–2148, 2003. 30

[94] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *Signal Processing Magazine, IEEE*, 22(4):54–69, 2005. 21, 30, 95

[95] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 1–14. ACM, 2007. 42, 43

[96] A. Polacco and K. Backes. The amazon go concept: Implications, applications, and sustainability. *Journal of Business & Management*, 24(1), 2018. 3

[97] N. B. Priyantha. *The cricket indoor location system*. PhD thesis, Massachusetts Institute of Technology, 2005. 43

[98] M. Quigley, D. Stavens, A. Coates, and S. Thrun. Sub-meter indoor localization in unmodified environments with inexpensive sensors. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2039–2046. IEEE, 2010. 84

[99] M. Z. Rahman. Beyond trilateration: Gps positioning geometry and analytical accuracy. *Global Navigation Satellite Systems: Signal, Theory and Applications*, pages 241–256, 2012. 41

[100] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 293–304. ACM, 2012. 2

[101] N. Rajagopal, P. Lazik, and A. Rowe. Hybrid visible light communication for cameras and low-power embedded devices. In *Proceedings of the 1st ACM MobiCom workshop on Visible light communication systems*, pages 33–38. ACM, 2014. 2

[102] N. Rajagopal, P. Lazik, and A. Rowe. Visual light landmarks for mobile devices. In *Proceedings of the 13th international symposium on Information processing in sensor networks*, pages 249–260. IEEE Press, 2014. 2

[103] N. Rajagopal, S. Chayapathy, B. Sinopoli, and A. Rowe. Beacon placement for range-based indoor localization. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2016. 5, 19

[104] N. Rajagopal, P. Lazik, N. Pereira, S. Chayapathy, B. Sinopoli, and A. Rowe. Enhancing indoor smartphone location acquisition using floor plans. In *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pages 278–289. IEEE Press, 2018. 7, 14, 18, 40

[105] N. Rajagopal, J. Miller, K. R. Kumar, A. Luong, and A. Rowe. Improving augmented reality relocalization using beacons and magnetic field maps. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2019. 7, 8, 67, 82

[106] T. Rajapaksha, X. Qiu, E. Cheng, and I. Burnett. Geometrical room geometry estimation from room impulse responses. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 331–335. IEEE, 2016. 66

[107] A. Ranganathan. *Physical-layer techniques for secure proximity verification and localization*. PhD thesis, ETH Zurich, 2016. 4

[108] M. A. Richards. *Fundamentals of radar signal processing*. Tata McGraw-Hill Education, 2005. 3

[109] M. A. Richards, J. Scheer, W. A. Holm, and W. L. Melvin. *Principles of modern radar*. Citeseer, 2010. 3

[110] J. O. Roa, A. R. Jiménez, F. Seco, J. C. Prieto, and J. Ealo. Optimal placement of sensors for trilateration: regular lattices vs meta-heuristic solutions. In *Computer Aided Systems Theory–EUROCAST 2007*, pages 780–787. Springer, 2007. 22

[111] P. Robertson, M. Frassl, M. Angermann, M. Doniec, B. J. Julian, M. G. Puyol, M. Khider, M. Lichtenstern, and L. Bruno. Simultaneous localization and mapping for pedestrians using distortions of the local magnetic field intensity in large indoor environments. In *International conference on indoor positioning and indoor navigation*, pages 1–10. IEEE, 2013. 83

[112] M. Rudafshani and S. Datta. Localization in wireless sensor networks. In *2007 6th International Symposium on Information Processing in Sensor Networks*, pages 51–60. IEEE, 2007. 65

[113] N. Salman, H. K. Maheshwari, A. H. Kemp, and M. Ghogho. Effects of anchor placement

on mean-crb for localization. In *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*, pages 115–118. IEEE, 2011. 21

[114] A. Savvides, H. Park, and M. B. Srivastava. The n-hop multilateration primitive for node localization problems. *Mobile Networks and Applications*, 8(4):443–451, 2003. 41

[115] Y. Shen, S. Mazuelas, and M. Z. Win. Network navigation: Theory and interpretation. *IEEE Journal on Selected Areas in Communications*, 30(9):1823–1834, 2012. 11

[116] O. Shih and A. Rowe. Can a phone hear the shape of a room? In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, pages 277–288. ACM, 2019. 66

[117] Y. Shu, C. Bo, G. Shen, C. Zhao, L. Li, and F. Zhao. Magicol: Indoor localization using pervasive magnetic field and opportunistic wifi sensing. *IEEE Journal on Selected Areas in Communications*, 33(7):1443–1457, 2015. 83

[118] B. Silva and G. P. Hancke. Practical challenges of ir-uwb based ranging in harsh industrial environments. In *Industrial Informatics (INDIN), 2015 IEEE 13th International Conference on*, pages 618–623. IEEE, 2015. 58

[119] B. Silva and G. P. Hancke. Ir-uwb-based non-line-of-sight identification in harsh environments: Principles and challenges. *IEEE Transactions on Industrial Informatics*, 12(3): 1188–1195, 2016. 42

[120] J. Smith and J. Abel. Closed-form least-squares source location estimation from range-difference measurements. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(12):1661–1669, 1987. 41

[121] M. Spirito et al. On the accuracy of cellular mobile station location estimation. *Vehicular Technology, IEEE Transactions on*, 50(3):674–685, 2001. 21, 30, 95

[122] R. Stansifer. Exact solution of a three dimensional hyperbolic positioning system. *From Florida Institute of Technology Research Paper*, 2011. 41

[123] K. P. Subbu, B. Gozick, and R. Dantu. Locateme: Magnetic-fields-based indoor localization using smartphones. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(4):73, 2013. 83

[124] D. Sun, A. Kleiner, and T. M. Wendt. Multi-robot range-only slam by active sensor nodes for urban search and rescue. In *Robot Soccer World Cup*, pages 318–330. Springer, 2008. 66

[125] H. Surmann, A. Nüchter, and J. Hertzberg. An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45(3-4):181–198, 2003. 66

[126] T. Taketomi, H. Uchiyama, and S. Ikeda. Visual slam algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):16, 2017. 66

[127] A. Tayebi, J. Gomez Perez, F. M. S. d. Adana Herrero, and O. Gutierrez Blanco. The application of ray-tracing to mobile localization using the direction of arrival and received signal strength in multipath indoor environments. *Progress In Electromagnetics Research*, 91:1–15, 2009. 44

[128] O. Tekdas and V. Isler. Sensor placement for triangulation-based localization. *Automation Science and Engineering, IEEE Transactions on*, 7(3):681–685, 2010. 21, 95

[129] S. Tervo and T. Tossavainen. 3d room geometry estimation from measured impulse responses. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 513–516. IEEE, 2012. 66

[130] C. Thomson, G. Apostolopoulos, D. Backes, and J. Boehm. Mobile laser scanning for indoor modelling. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, 5:W2, 2013. 66

[131] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. 2005. 66, 71, 72

[132] S. Thrun et al. Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, 1(1-35):1. 66

[133] D. Vasisht, S. Kumar, and D. Katabi. Decimeter-level localization with a single wifi access point. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, pages 165–178, 2016. 2

[134] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 197–210. ACM, 2012. 83

[135] Q. Wang, H. Luo, F. Zhao, and W. Shao. An indoor self-localization algorithm using the calibration of the online magnetic fingerprints and indoor landmarks. In *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*, pages 1–8. IEEE, 2016. 83

[136] S. Wang, H. Wen, R. Clark, and N. Trigoni. Keyframe based large-scale indoor localisation using geomagnetic field and motion pattern. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 1910–1917. IEEE, 2016. 83

[137] X. Wang, Z. Wang, and B. O'Dea. A toa-based location algorithm reducing the errors due to non-line-of-sight (nlos) propagation. *IEEE Transactions on Vehicular Technology*, 52 (1):112–116, 2003. 25

[138] Y.-T. Wang, R. Zheng, and D. Zhao. Towards zero-configuration indoor localization using asynchronous acoustic beacons. In *2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES)*, pages 32–39. IEEE, 2016. 43

[139] J. Wilson and N. Patwari. Radio tomographic imaging with wireless networks. *IEEE Transactions on Mobile Computing*, 9(5):621–632, 2010. 3

[140] Y. I. Wu, H. Wang, and X. Zheng. Wsn localization using rss in three-dimensional space;a geometric method with closed-form solution. *IEEE Sensors Journal*, 16(11):4397–4404, June 2016. ISSN 1530-437X. doi: 10.1109/JSEN.2016.2547444. 42

[141] Z. Xiao, H. Wen, A. Markham, and N. Trigoni. Lightweight map matching for indoor localisation using conditional random fields. In *Information Processing in Sensor Networks, IPSN-14 Proceedings of the 13th International Symposium on*, pages 131–142.

IEEE, 2014. 2, 44

[142] Z. Yang and Y. Liu. Quality of trilateration: Confidence-based iterative localization. *IEEE Transactions on Parallel and Distributed Systems*, 21(5):631–640, 2009. 22

[143] G. Younes, D. Asmar, E. Shammas, and J. Zelek. Keyframe-based monocular slam: design, survey, and future directions. *Robotics and Autonomous Systems*, 98:67–88, 2017. 66

[144] M. Youssef, M. Mah, and A. Agrawala. Challenges: device-free passive localization for wireless environments. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 222–229. ACM, 2007. 3

[145] P. Zhao, C. X. Lu, J. Wang, C. Chen, W. Wang, N. Trigoni, and A. Markham. mid: Tracking and identifying people with millimeter wave radar. In *International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2019. 3

[146] P. Zhou, M. Li, and G. Shen. Use it free: Instantly knowing your phone attitude. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 605–616. ACM, 2014. 67, 83

[147] Y. Zhou. An efficient least-squares trilateration algorithm for mobile robot localization. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3474–3479. IEEE, 2009. 42

[148] Y. Zhou. A closed-form algorithm for the least-squares trilateration problem. *Robotica*, 29(3):375–389, 2011. 42