WIRELESS SENSING: MATERIAL IDENTIFICATION AND LOCALIZATION

BY

ASHUTOSH MAKRAND DHEKNE

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Doctoral Committee:

Professor Romit Roy Choudhury, Chair
Professor Klara Nahrstedt
Associate Professor Philip Brighten Godfrey
Dr. Karthikeyan Sundaresan, NEC Labs
Associate Professor Nicholas Lane, University of Oxford

ABSTRACT

Wireless signals are everywhere around us, and they have truly revolutionized the world by all standards. When one thinks of this revolution, one envisions the advances in wireless communication—TV broadcasts, FM radios, WiFi, Bluetooth, cellular mobile phones, and even wireless chips inside the human body. What gets less appreciated, however, is that wireless signals can also be a powerful sensor. The fact that wireless signals touch and penetrate all objects in our environment, and bounce back, make them a powerful lens to view our world through.

This thesis focuses on using wireless signals as sensors. We will explore how modifications to wireless signal propagation can reveal the physical properties of the materials that these signals have passed through. This enables identification of materials without touching them or performing any chemical analysis on them. We will show the ability to distinguish between closely related liquids, such as Pepsi and Coca-Cola, or distilled water and mineral water, by simply passing wireless signals through the liquids, and analyzing the signals that emerge on the other side.

The propagation delay of wireless signals when passing through air can reveal the distance between a transmitter and a receiver. We show how this primitive can be extended for localization with applications to sports, battlefields, and emergency response. Through modifications to the distance measurement mechanisms, we show how localization is possible even when wireless devices are constantly under motion.

We end by discussing future directions in which both of these sensing techniques can be extended. Under the right conditions, it might be possible to localize an object to 5mm precision with applications in robotic machines, augmented reality, and virtual reality. We then discuss the possibility of using reflections of wireless signals, for example, to determine soil moisture content in agricultural fields.

To Neha, for her support and encouragement.

ACKNOWLEDGMENTS

I am grateful to the University of Illinois at Urbana Champaign (UIUC) for the opportunity of pursuing my Ph.D. here. It has been a fantastic time and the campus has nurtured me creating a home away from home. I have enjoyed the colorful falls, the serene winters, the cheerful spring blossoms, and the welcoming brick buildings at UIUC. I am grateful to have attended this esteemed institution.

None of this work would have come to fruition without the constant encouragement, guidance, and support from my advisor, Professor Romit Roy Choudhury. I will remain forever indebted to him for inculcating in me, the values of scientific inquiry, critical thinking, and sound validation. His words of advice, caution, and encouragement, will continue to resonate in my ears for the rest of my life. Thank you, Romit.

I am also grateful to my Ph.D. committee members and mentors—Professor Klara Nahrstedt, Professor Brighten Godfrey, Dr. Karthikeyan Sundaresan, and Professor Nicholas Lane. Their support and insights have helped shape this thesis. I would like to thank Klara for her suggestions and insights in the DroneNet project and supporting the use of Intelligent Robotics Lab for the cricket project. I worked closely with Karthik during my internship at NEC Labs. Thank you, Karthik, for the thorough technical discussions on the TrackIO project and for dedicating so much time for it. I would also like to thank Ayon from NEC Labs for his mentorship. His attention to detail, hard work, and meticulous execution of projects is commendable and I am fortunate to have him as a mentor.

I wish to extend many thanks to Mahanth Gowda, who has been an excellent friend, an inspiration, and a co-author in my Ph.D. journey. I have learned so much from him. His dedication to science, sharp eye at spotting errors, and a peaceful soft-spoken persona has made it a pleasure to work with him over the years. I will always cherish our discussions on everything under the sun (and beyond).

It has been great to work with Sheng Shen on one of the most intricate projects that I have undertaken. He has been my go-to person for all things related to inertial sensors since then. Working with Sheng has always been frictionless, be it on a project or as a co-TA. Special thanks to Nirupam Roy for hours of interactions vetting crazy ideas, from picture passwords to interrelations between RF and acoustics. I also appreciate his insightful guidance during the last phase of my Ph.D. It has been amazing to work with Professor Haitham Hassanieh on the liquid identification work and I appreciate his guidance, eye for detail, and enthusiasm. I thank the whole SyNRG group for playing an enthusiastic and active role in all of my works. I could not have asked for a better team.

I wish to thank the Computer Science Department—faculty, and the administrative staff—for an amazing learning experience, support through multiple teaching assistantships, and guidance throughout the Ph.D. process. Also, thanks to the Electrical and Computer Engineering Department for multiple teaching assistantships.

Everything that I am and everything that I am able to do is due to the unrelenting support of my family. Thanks a ton to my wife, Neha, for being there with me—through the ups and downs of the drones, buoying me up through the liquid season, and through the vanishing acts I have been up to during these years. Thanks for your patience, confidence, support, and love. This thesis is dedicated to you. Thanks to Om for being such an adorable kid and being patient with me all this time. His hugs infuse new energy into me every single day. Love you, Om. Thanks to my parents, Makrand and Suhasini, for cradling my curiosity from a young age. None of this was possible without their nurturing and unwavering confidence in me. Thanks also to Neha's parents, Uday and Hema, for their constant support and blessings in all my endeavors. Thanks for your love, your keen interest in my projects, and for believing in me. Thanks to my little sister, Mrudula, for her encouragement all these years. I have many fond childhood memories with her.

Innumerable others have been part of this journey and I thank all them for their support, encouragement, and confidence in me. Thanks for an enriching Ph.D. experience.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

Wireless communication has become ubiquitous in recent years with almost half of Internet traffic being generated from a mobile device. This has led to advancements in wireless communication which focus on packing higher amounts of information into the wireless channel. As the wireless signals travel from one device to another, the environment causes distortions in these signals. In general, these distortions corrupt the transmitted signal limiting the amount of information that can be packed into wireless signals. It is therefore a nemesis for the wireless engineer. For us, however, these distortions are of great interest. Since these distortions are caused by the impact of the objects and obstacles that the wireless signals traverse, a lot of information about the environment is imbibed into the signals. Carefully extracting this information could help us sense the environment.

In this work, we study propagation of wireless signals as it pertains to two sensing applications: (1) localization, and, (2) material identification. Localization is the ability to estimate the location of an object or person in some frame of reference. Distance measurement provides a primitive necessary to estimate location. If we measure the time wireless signals require to travel from a sender to a receiver, we can calculate the distance between the two devices, by simply multiplying the time with speed of light. When a device's distance is measured from three or more fixed known locations, called anchor points, we can geometrically solve for its location. This is commonly known as trilateration. Relaxing from this traditional requirement of *three, static* anchors, we enable wireless localization in more restricted settings: (1) tracking a fast moving ball in the game of cricket [1] where only two anchors are available, (2) tracking a fast-moving team of players [2] without any fixed anchors on the ground, and, (3) tracking a team of emergency-responders indoors using a moving anchor outside the building. We track the cricket ball by adding angle-of-arrival to the available ranging information. We encounter issues of dilution of precision and human body occlusions when tracking players, which we solve through a dynamic designation of some players as anchors.

Most localization applications usually assume that the signals are traveling through air and hence are able to multiply the travel time by the speed of light. Instead, if the wireless signal travels through a medium other than air, it travels slower due to the different refractive index of that material. Further, a part of the wireless signal's energy is absorbed by the medium. We observe these effects on a wireless signal and estimate the material's electrical properties enabling material identification [3]. Of course, the slowdown in various materials due to refractive index can be minuscule, to the tune of a few picoseconds. We show how splitting the transmitted signal to pass over a reference path can enable precise measurements to identify material properties. Testing our experimental setup with liquids, we can distinguish between, for example, Pepsi and Coca-Cola, vegetable oil and motor lubricant, and even distilled water and mineral water. Such a capability on a user's smart phone can have applications in ascertaining food quality, detecting adulteration of drinks, and also in airport security.

This thesis is organized as follows. We first present in detail, the work on material identification using wireless signals. In the rest of this thesis, we will first present a primer on wireless ranging and localization, introduce the current state-of-the-art algorithms for localization, and present the hardware used in this work. We will then progressively relax assumptions about the anchors and show how the challenges introduced by these relaxations can be overcome through algorithmic, architectural, and protocol modifications.

CHAPTER 2: IDENTIFYING MATERIALS USING WIRELESS SIGNALS

2.1 INTRODUCTION

Liquids are known to be identifiable by their *permittivity*. At a high level, permittivity is the resistance that a liquid offers when an electrical field is formed inside it. Permittivity, however, is very difficult to measure. Today, devices that estimate the permittivity of a liquid rely on optical spectroscopy. These are large bulky equipment that cost in the excess of $\$40000$ and hence are limited to laboratory settings [4]. Smaller devices exist at $\approx$ $\$18000$, but sacrifice accuracy and require dipping a probe into the liquid [5]. In this work, we ask *whether it is feasible to estimate permittivity and identify liquids by non-invasively shining wireless signals from cheap off-the-self radios.* If successful, the applications could be many, ranging from airport security on liquids, quality control on stored drinks/wines in warehouses, cheap detection of water contamination especially in countries with limited access to clean water, all the way to futuristic cups that can measure the calorie count of a drink inside the cup, and devices that can analyze the blood content. This work takes the initial small step towards this vision by developing the core capability first i.e., a reliable wireless liquid identifier.

We present *LiquID*, a cheap lightweight system that uses wireless signals to identify liquids. LiquID uses two cheap off-the-self ultra-wideband (UWB) radios [6]. It shines an UWB signal from one side of a liquid container and receives the signal on the other side. LiquID then analyzes the characteristics of the received signals to estimate the permittivity and identify the liquid. In designing LiquID, we build on several past papers that use wireless signals to classify liquids [7, 8, 9, 10]. The most recent of these papers [10] demonstrates the ability to classify 10 liquids using RFID readers and tags. However, the solution applies only to a specific set of 10 liquids that exhibit special wavelength relationships to each other. In contrast, LiquID estimates the intrinsic permittivity of any given liquid. Our experiments span over 33 different liquids, essentially demonstrating a general solution to liquid identification.

LiquID uses UWB signals with a $4GHz$ center frequency and a $1GHz$ bandwidth to measure two key attributes of liquids: *refractive index* and *loss factor*. These measurable attributes ultimately map to permittivity. When the UWB signal passes through the liquid, the liquid interacts with the signal in the two following ways:

(1) It slows down the signal's propagation speed. The slow down is referred to as *refrac-*

tive index which in turn reduces the wavelength of the signal (since speed of propagation $v = f\lambda$). This manifests as a change in the phase of the received signal.

(2) The liquid also attenuates the signal's amplitude. The attenuation is called *loss factor* and translates to a weaker received signal strength (RSS or RSSI).

By measuring the phase and the RSSI of the signal at the receiver, we had expected to compute the above two attributes. Unfortunately, since the phase wraps multiple times (around 2π) inside the liquid, the measured phase is not a unique signature[1]. However, if the absolute propagation delay can also be estimated alongside the phase, perhaps the wrapped-phase can be unwrapped. Now, given that wireless propagation delay is in the granularity of picoseconds, precise estimation is a challenge with today's UWB receivers. Even clock synchronization is difficult to achieve to this consistent precision [6]. Moreover, noise, multipath, and the plastic material (of the container), all affect the phase and RSSI, polluting the measurements drastically. Extracting out the signal slowdown and the attenuation, in face of all these complications, forms the root of challenges in liquid identification.

LiquID addresses these challenges through successive stages of signal analysis and engineering. Two key ideas underpin the solution.

(1) By using a simple wire connected between the transmitter and receiver, LiquID develops a reference frame for time. In other words, the wireless transmission is forked between the wire and the wireless antenna, so that the receiver can use the wired signal as the baseline. This obviates the need for clock-synchronization, since measurements through the liquid, as well as through the air, can both be normalized over the same denominator. Thus, by careful "double-differencing" techniques against wire and air, the liquid's *relative* time of flight (ToF), phase, and RSSI can be measured with consistent accuracy. Since permittivity, refractive index, and loss factors are all defined relative to vacuum, such relative measurements align even better with the final requirements.

(2) The second opportunity arrives from the observation that phase can be derived from estimated ToFs, as well as from direct measurements. This serves as redundant information, which in turn can be utilized for coping with noise. Thus, LiquID will use the modeled phase (from ToFs) to roughly estimate the integer ambiguity, then adjust the ambiguity from the measured phase, and finally refine the ToF from the precise ambiguity.

[1] This is analogous to measuring the distance between point A and B with only the least significant digit of the measurement, say 6. We would not be able to tell if the actual distance was 6, 16, 26, 36, and so on. Measured phase is analogous to this least significant digit.

This is a form of jointly estimating phase and ToF to arrive at the signal slowdown in liquid.

With these techniques, and subsequent stages of channel interpolation, container compensation, RSSI modeling, and others, LiquID arrives at an estimate of permittivity. The estimates are validated against specialized vector network analyzer (VNA) equipments (fortunately accessible in a Microwave Communications Lab).

Figure 2.1 shows our experimentation platform, with UWB antennas on two sides of the $2cm$ wide plastic container. The transmitter and receiver are not clock synchronized, neither do they need to be placed at any specific distance from the liquid. Regular multipath environments are acceptable, and any liquid can be poured. Our only requirement is that the liquid container be placed perpendicular to the wireless link, so that the signals are not incident obliquely onto the vertical cross-section of the liquid. This is important because oblique incidence triggers complex bending of signals at the boundary of the liquid, “smudging” the precision of our measurements. Of course, no special alignment is needed to ensure perpendicularity; just rough manual placement is more than adequate.

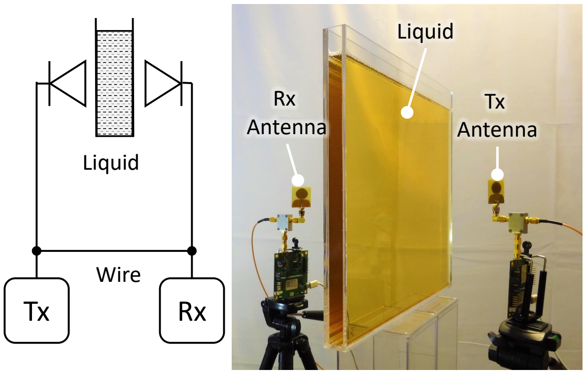


Figure 2.1: LiquID experimental setup: (a) The UWB radios on two sides of the liquid-filled container placed perpendicularly. (b) The system schematic, with signals transmitted both through the antenna and wire.

We measure 33 different liquids, their permittivity ranging from the minimum $P_{air} = (1 + j0)$ to the maximum of $P_{water} = (75.7 + j14.3)$ (note that permittivity is a complex number). The 33 liquids cover the entire permittivity spectrum, with examples as: saline water, orange juice, pepsi, coke, isopropyl alcohol, ethanol, green tea, sweet tea, peanut oil, canola oil, etc. Median permittivity error is 9%, offering a resolution to separate sweet tea from green tea, orange juice from apple juice, olive oil from peanut oil, etc. We believe our permittivity errors are promising in comparison to specialized equipments (costing $\$134000$) that quote 5% error using invasive probes dipped in the liquid. Our system, on the other hand, costs $\$150$, is non-invasive, and is practical for various applications.

In sum, this work's contributions may be summarized as follows:

- *Identifying the viability of extracting material permittivity from the properties of signals passing through the material.* While we demonstrate our systems for liquids, the techniques generalize to solids as well (with suitable modifications to the Physics models).
- *Developing synchronization-free techniques that quantify a signal's slow-down (inside a liquid) with high precision.* Building a robust system that does not require training or calibration, but identifies liquids with consistency.

We elaborate on these contributions next, starting with some background on permittivity, followed by overview, system design, and evaluation.

2.2 BACKGROUND ON PERMITTIVITY

The permittivity of a given liquid is technically a complex number ϵ^* as follows:

$$\epsilon^* = \epsilon' - j\epsilon'' \tag{2.1}$$

Here, ϵ' is the *dielectric constant* and ϵ'' is the *loss factor* of the liquid [11]. For vacuum, the loss factor is zero, therefore its permittivity is a real number denoted as $\epsilon_0 = \epsilon'_{vac}$. By convention, a liquid's permittivity is expressed relative to vacuum, as:

$$\epsilon^* = \left(\frac{\epsilon'}{\epsilon_0}\right) - j\left(\frac{\epsilon''}{\epsilon_0}\right) \tag{2.2}$$

Our final goal in this work is to estimate these relative values denoted as $\epsilon'_r = \left(\frac{\epsilon'}{\epsilon_0}\right)$ and $\epsilon''_r = \left(\frac{\epsilon''}{\epsilon_0}\right)$. However, since neither of them can be measured directly, we have to rely on two other measurable quantities – *refractive index* and *attenuation* – to indirectly arrive at this estimation. To understand these indirect relationships, let us briefly look at refractive index and attenuation first.

2.2.1 Refractive Index (RI)

The refractive index n of a material is the ratio of the speed of light in vacuum, c, to the speed of electromagnetic waves in that material, v.

$$n = \frac{c}{v} \tag{2.3}$$

Even though a wave slows down in a given material, its frequency cannot change. As a result, the wave experiences a decrease in wavelength [12], dictated by:

$$f = \frac{c}{\lambda_0} = \frac{v}{\lambda} \tag{2.4}$$

where λ_0 is the wavelength of electromagnetic waves in vacuum and λ is the wavelength in that material. Now, refractive index is related to complex permittivity [11] as:

$$n = \frac{c}{v} = \frac{\lambda_0}{\lambda} = \sqrt{\frac{1}{2}\epsilon_r' \left\{ \sqrt{1 + \left(\frac{\epsilon_r''}{\epsilon_r'}\right)^2} + 1 \right\}} \tag{2.5}$$

This implies that refractive index alone is insufficient to estimate the 2 unknowns ϵ_r' and ϵ_r''. We need another equation to solve for complex permittivity.

2.2.2 Attenuation Factor (AF)

The "attenuation factor" of a material is defined as the width of the material needed to decay the strength of the electromagnetic field to $1/e = 0.368$ of its original (incident) value [13]. This width, α_d, is given by:

$$\alpha_d = \frac{\lambda_0}{2\pi} \sqrt{\frac{2}{\epsilon_r'(\sqrt{1 + \left(\frac{\epsilon_r''}{\epsilon_r'}\right)^2} - 1)}} \tag{2.6}$$

For a liquid width of $d > \alpha_d$, the decay will be larger than $1/e$, and vice versa. Thus, the signal strength, SS, at the exiting boundary will be related to that at the entering boundary by:

$$\frac{SS_{exit}}{SS_{entry}} = \left(\frac{1}{e}\right)^{\frac{d}{\alpha_d}} \tag{2.7}$$

Measuring the LHS and knowing the d are adequate to obtain α_d. However, measuring the SS at precise liquid boundaries is not practical (since the antenna cannot be placed at the exact air-liquid interface). One potential solution is to measure the SS away from the boundary, and compute the reduction with and without liquid in the container. Unfortunately, this approach fails since signals also undergo reflections at liquid boundaries (as shown by gray arrows in Figure 2.2) and only a smaller fraction passes through the liquid. Unless we model these reflections, our estimates will be incorrect. We discuss this next and then arrive at an unified equation to obtain the final liquid permittivity.

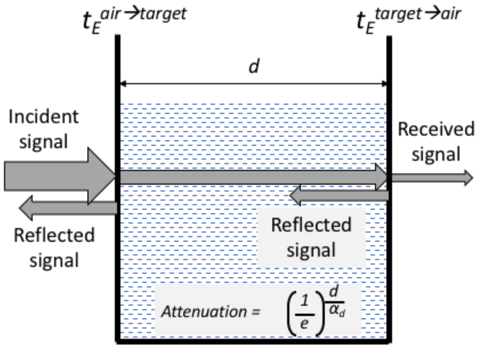

$t_E^{air\rightarrow target}$
$t_E^{target\rightarrow air}$
d
Incident signal
Received signal
Reflected signal
Reflected signal
Attenuation = $\left(\frac{1}{e}\right)^{\frac{d}{\alpha_d}}$

Figure 2.2: Net signal attenuation after passing through liquid caused by (1) reflections at liquid boundaries, and (2) attenuation factor of the liquid.

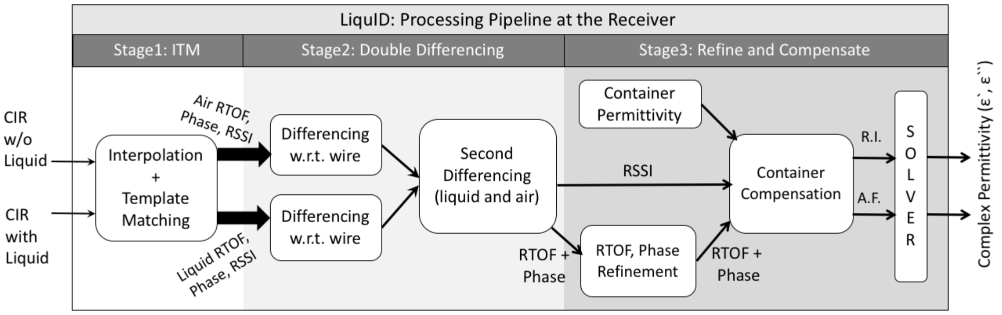


Figure 2.3: Four stages of processing, with Decawave's CIRs as inputs and liquid permittivity as output.

2.2.3 Transmission Coefficient

When a traveling wave encounters a material boundary, only a portion of the incident energy penetrates into the new medium; the rest is reflected back. The fraction of penetrated energy is given by the transmission coefficient, t_E:

$$t_E = \frac{2Z_2}{Z_2 + Z_1} \tag{2.8}$$

where Z_1 is the intrinsic impedance of the material the waves are entering from, and Z_2 is the intrinsic impedance of the material the waves are entering into. Fortunately, intrinsic impedance is also related to permittivity as follows:

$$Z^* = \frac{Z_0}{\sqrt{\epsilon^*}} \tag{2.9}$$

where Z_0 is the impedance of free space, and ϵ^* is the permittivity of the entering material. Thus, it is now possible to combine all factors.

For this, we denote the transmission coefficient at the air-target boundary as $t_E^{air \to target}$ and that between the target-air boundary as $t_E^{target \to air}$. Then, we combine Equation 2.6 and Equation 2.8 to obtain the complete expression for attenuation of the signal before it reaches the receiver. The received signal strength (RSS) in the presence of the target liquid, RSS_{target}, compared to the RSS without the target, RSS_{air} is given by:

$$\frac{RSS_{target}}{RSS_{air}} = t_E^{air \to target} \cdot t_E^{target \to air} \cdot \left(\frac{1}{e}\right)^{\frac{d}{\alpha_d}} \tag{2.10}$$

Since Equation 2.10 also depends on both ϵ' and ϵ'' we solve it together with Equation 2.5 to obtain the complex permittivity of the target liquid, for a known liquid width of d. With this background, we are now ready to describe the LiquID system.

2.3 SYSTEM OVERVIEW

Figure 2.3 shows the computational pipeline underlying LiquID. We present an overview of the whole system first, followed by technical details of each stage.

Recall from Figure 2.1 that our system set-up is essentially a plastic container (to hold the liquid) with two UWB radios on either side. The radios are also connected with a wire to serve as a latency reference (explained later). The transmitter (Tx) forks the signal over the wire and the wireless antenna; the receiver (Rx) receives both over separate channels. To estimate permittivity, we perform 2 transmissions, first through a liquid-filled container, and then without the container (i.e., only air between the Tx and Rx). For each measurement, the Decawave receiver provides a discrete *channel impulse response* (CIR) shown in Figure 2.4.

Given UWB's $1GHz$ wide bandwidth, the CIR taps (on the X-axis) are at 1 nanosecond time gaps. Nonetheless, Decawave performs fractional optimizations [14] and brings the ToF estimation down to ≈ 350 picoseconds. This implies that a distance computation using ToF would result in around $10cm$ error. While promising for applications like localization,

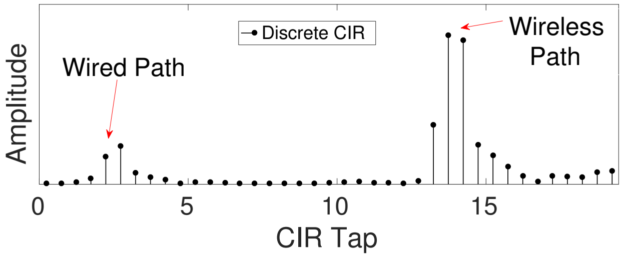


Figure 2.4: A sample discrete CIR obtained directly from the Decawave UWB platform.

separating liquids based on signal slowdown requires ≈ 30 picosecond resolution. This motivates the core problem of significantly improving ToF, in addition to carefully refining RSSI to finally achieve permittivity.

The LiquID receiver begins by accepting 2 CIRs as inputs – one through liquid and one through air. The processing on both the CIRs are identical, so let us consider only the CIR through liquid, denoted CIR_{liq}. Figure 2.4 shows an example CIR_{liq} – the first peak indicates the arrival of the wired signal while the second peak is due to the wireless signal (through the liquid).

In **Stage 1**, the goal is to identify the correct time at which the line of sight (LOS) signal arrives at the receiver. Given that Decawave's nanosecond resolution is inadequate, LiquID first performs a frequency-domain interpolation on CIR_{liq}. This inserts multiple samples between adjacent CIR taps, while also interpolating the phase of the samples. The next task is to identify the correct sample (in the interpolated CIR) corresponding to the LOS arrival. For this, LiquID extracts the portion of CIR corresponding to the wire – which captures the UWB hardware distortions – and correlates this CIR_{wire} against CIR_{liq}. The (sub-nanosecond) sample that spikes in correlation is denoted T_{liq}, and is declared as the time of LOS arrival through liquid. The time of arrival for the wired signal is denoted T_{wire}, and was easier to detect since it was free of multipath.

Stage 2: To translate T_{liq} to absolute time of flight (ToF), the Rx must be precisely clock synchronized with the Tx. Although Decawave's synchronization is sophisticated, it falls short of the needed 30 picosecond resolution. To completely sidestep synchronization, LiquID estimates *relative ToF* (RTOF) with respect to the wired path. The idea is that ToF through the wire remains very stable, hence $(T_{liq} - T_{wire})$ is essentially the relative ToF of the signal through the liquid. Similarly, $(T_{air} - T_{wire})$ is the relative ToF through air. Now, "differencing" again between the two quantities, (i.e., $(T_{liq} - T_{wire}) - (T_{air} - T_{wire})$) yields the relative ToF of liquid with respect to air (i.e., $(T_{liq} - T_{air})$). This, by design, is free of clock synchronization, and more importantly, captures the ToF *only* through the width of the liquid column (see Fig. 2.9).

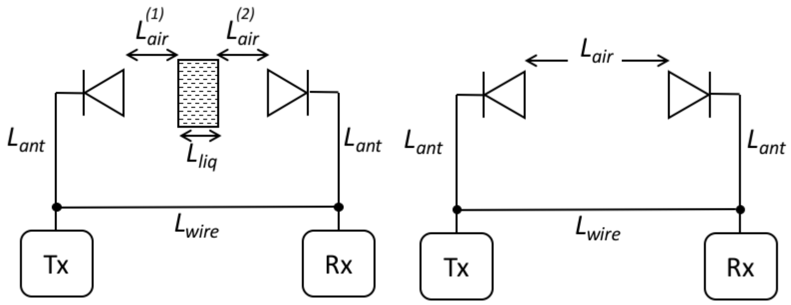


Figure 2.5: The ToF consists of various delay components in addition to delay through (a) liquid or (b) air alone.

Stage 3 focuses on refining ToF by leveraging signal phase. Observe that phase can be obtained from two sources: (1) direct measurement from the Rx, and (2) derived by dividing RTOF by λ, mod 2π. Of course, phase is affected by *integer ambiguity*, q (i.e., the signal may have traveled q full wavelengths during its time of flight). Thus, LiquID uses the (RTOF + *derived* phase) to obtain an estimate of q, refines q using the *measured* phase, and then refines relative ToF (RTOF) using the refined q. In a separate thread, LiquID also extracts the RSSI of the received signals, and again via "differencing", computes the relative amplitude attenuation of liquid over air (see Fig. 2.13). However, to converge on actual permittivity, we still need to compensate the plastic container's impact on RSSI.

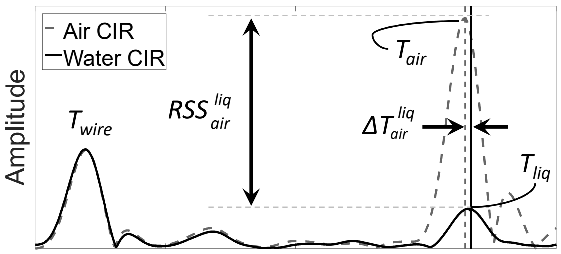


Figure 2.6: The wired reference aligns T_{wire} in the CIR, thereby enabling relative ToF and relative RSS between liquid and air.

Thus, as a one-time measurement, LiquID sends a signal through the empty container, computes the refined ToF, phase, and RSSI, and ultimately derives the permittivity of the container material. The container's permittivity is fed into the processing pipeline so that the UWB signal's behavior at the boundary of plastic and liquid can be modeled and compensated. The output from this "container compensation" module are the *refractive index (RI)* and the *attenuation factor (AF)* of the liquid, which are then fed into a MATLAB solver. This yields the final ϵ' and ϵ'', and hence permittivity.

2.4.1 Stage1: Interpolation + Template Matching

A *channel impulse response* (CIR) describes the delays and strengths of various signal paths between Tx and the Rx. Figure 2.4 shows the CIR_{liq} from Decawave hardware – the resolution of the X-axis, a function of bandwidth, is 1 nanosecond [14]. The first task is to identify the time instant that corresponds to the arrival of the line of sight (LOS) signal path from the Tx to the Rx. Of course, this time may be somewhere between the available CIR samples (i.e., our CIR is only a measurement at the nanosecond granularity but the LOS signal could arrive at any time). This calls for a higher resolution CIR.

Interpolation: For higher resolution, we up-sample the CIR by adding *zeros* between existing samples and then apply a 1GHz low pass filter. Figure 2.7 shows this interpolated CIR. As evident from this example, the highest point in the interpolated CIR is actually shifted from the highest point of the un-interpolated CIR. In other words, interpolation takes the CIR closer to its original analog form, allowing for better estimation of the LOS delay and amplitude[2]. This in turn allows for fractional (i.e., sub nanosecond) delay and more accurate amplitude estimation.

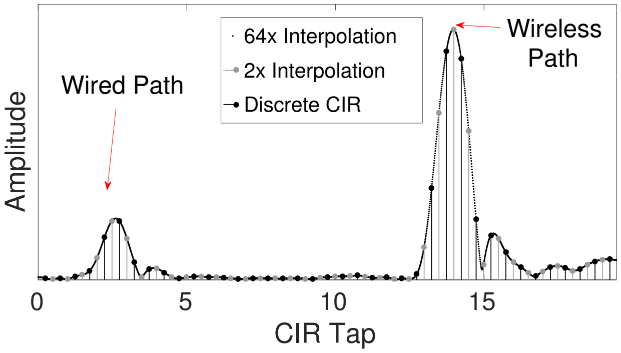


Figure 2.7: The interpolated version of a discrete CIR.

Template Matching: Of course, this estimation is still polluted by noise and various hardware distortions. Multipath, each adding a $Sinc()$ function around its time of arrival may also add to the pollution. The CIR we have is a sampled version of these aggregated $Sinc()s$ and hardware distortions. Thus, the peaks in our interpolated CIR may be spurious, i.e., the tallest sample may not correspond to the precise arrival time of the LOS signal.

[2] Observe that interpolation also maintains the complex nature of the CIR, offering correct phase at the intermediate points after interpolation.

To solve this problem, we utilize the CIR for the wired connection between the Tx and Rx. This CIR is free of multipath and captures the internal filtering effects (and other hardware distortions) of the UWB receiver. We call this the "template CIR" as shown in Figure 2.8 (and also subject it to interpolation). We then correlate the template CIR with the wireless CIR. The CIR sample at which the correlation spikes is declared, for now, as the *time of flight* (ToF) of the LOS signal.

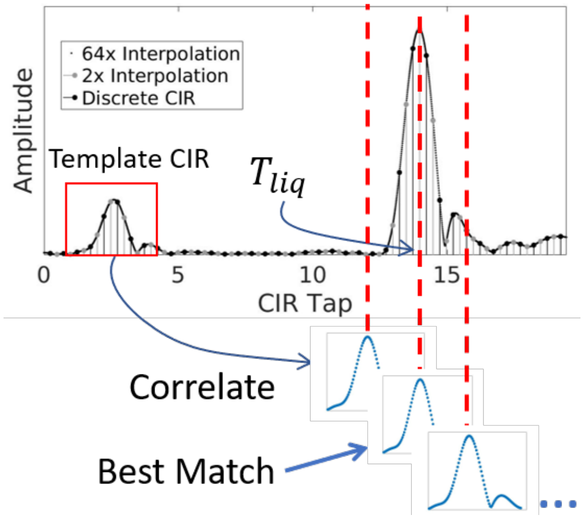


Figure 2.8: The CIR template from the wires channel used for correlating and finding the best sample, T_{liq}, at which LOS signal arrives through the liquid.

In summary, we transmit UWB signals through a liquid-filled container, as well as through air, each yielding one CIR. Both CIR_{liq} and CIR_{air} are first interpolated followed by template matching. After matching against CIR_{liq}, we denote the best-match sample for the wired path as T_{wire} and the one for the wireless LOS path, as T_{liq} (see Figure 2.8). Similarly, for CIR_{air}, we get T_{wire} and T_{air}. Indeed, T_{liq} and T_{air} are a crude estimate of the ToF, however, we expect to improve this by making them relative.

2.4.2 Stage 2: Double Differencing

If the transmitter and receiver were synchronized, the antenna-to-antenna time of flight (ToF) could be calculated from T_{liq} itself. However, achieving picoseconds level time synchronization is difficult even when using the same reference clock for two devices [15]. This motivates us to develop a synchronization-free technique. The simple opportunity is to utilize the wire delay (T_{wire}) as a fixed reference, and express the liquid or air path with respect to it. Specifically,

$$\begin{aligned} \Delta T_{wire}^{liq} &= T_{liq} - T_{wire} \\ \Delta T_{wire}^{air} &= T_{air} - T_{wire} \end{aligned} \tag{2.11}$$

While this obviates the need for synchronization, it is still inadequate for estimating ToF through liquid. The reason is that ΔT_{wire}^{liq} is *not just the signal propagation through the liquid column*, but also includes propagation delays through antenna connectors, air gaps between antennas and the liquid, and even through the material of the container. Figure 2.9(a) illustrates the path and we model this as:

$$\Delta T_{wire}^{liq} = \frac{2L_{ant}}{v_{wire}} + \frac{L_{air}^{(1)}}{c} + \frac{L_{liq}}{v} + \frac{L_{air}^{(2)}}{c} - \frac{L_{wire}}{v_{wire}} \tag{2.12}$$

where v is signal velocity in the liquid, L_{liq} is the liquid's column depth, and $L_{air}^{(1)}$ and $L_{air}^{(2)}$ are the first and second air gaps before and after the liquid column.

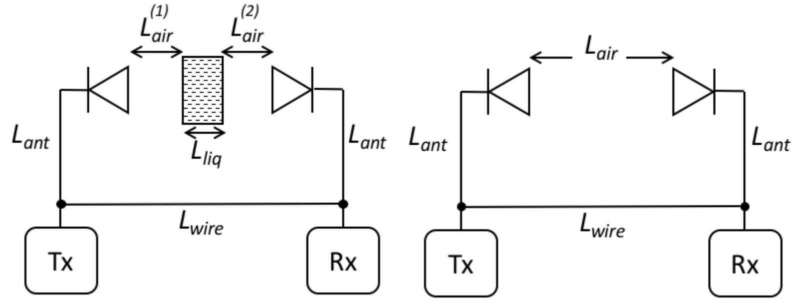


Figure 2.9: The ToF consists of various delay components in addition to delay through (a) liquid or (b) air alone.

Similarly, for ΔT_{wire}^{air}, shown in Figure 2.9(b), we can write a similar equation as:

$$\Delta T_{wire}^{air} = \frac{2L_{ant}}{v_{wire}} + \frac{L_{air}}{c} - \frac{L_{wire}}{v_{wire}} \tag{2.13}$$

where L_{ant} is the length of each antenna connector, L_{air} is the physical air gap between antennas, L_{wire} is length of the wired path, v_{wire} is speed of the signals in wire, and c is the speed of UWB signals in air.

To estimate the ToF *only* within the liquid column, we need to isolate out the term (L_{liq}/v) from Equation 2.12. To achieve this, we perform a second differencing operation between the above two equations as follows.

$$\begin{aligned}\Delta T_{air}^{liq} &= \Delta T_{wire}^{liq} - \Delta T_{wire}^{air} \\ \Delta T_{air}^{liq} &= \frac{L_{air}^{(1)}}{c} + \frac{L_{air}^{(2)}}{c} + \frac{L_{liq}}{v} - \frac{L_{air}}{c}\end{aligned} \tag{2.14}$$

Since $(\frac{L_{air}^{(1)}}{c} + \frac{L_{air}^{(2)}}{c} - \frac{L_{air}}{c}) = (-\frac{L_{liq}}{c})$, we have the ToF through the liquid, relative to that of air, as:

$$Relative\ ToF = \Delta T_{air}^{liq} = \frac{L_{liq}}{v} - \frac{L_{liq}}{c} \tag{2.15}$$

During an actual experiment, many values of ΔT_{wire}^{air} are obtained first (for about a minute) and then the liquid column is introduced. The median ΔT_{wire}^{air} is used for the differencing in Equation 2.14. Since the liquid column depth (L_{liq}) is accurately known, we solve Equation 2.15 using experimentally obtained values of ΔT_{air}^{liq} and estimate the velocity v of signals in the liquid. This v is essentially the slower signal velocity in liquid that we have been seeking to extract. We will improve this accuracy in the next stage and then plug the values in the LHS of Equation 2.5 to obtain refractive index.

Double Differencing Phase: Observe that the differencing operations hardly introduce errors in relative ToF; the root of ToF errors are still sourced in the original interpolation and template matching operations (i.e., in the estimation of T_{liq} and T_{air}). To refine this in stage 3 of the processing pipeline, we will use the phase corresponding to the samples T_{liq} and T_{air}, denoted ϕ_{liq} and ϕ_{air}, respectively. Phase brings value because it is an attribute of the original infinite bandwidth impulse that was obtained even before filtering at the receiver, hence, the phase under the CIR peaks are relatively constant as seen from Figure 2.10. The phase variations over the peak are about 1.5° in the wired path and around 7° in the air path. Therefore, inaccuracies in locating the exact peak do not significantly affect the phase, but can dramatically alter the ToF estimate. This is the key opportunity. However, to apply phase to relative ToF, we require similar double differencing operations to be imposed on phase as well.

Differencing phase is almost identical to ToF and can be expressed as follows.

$$\begin{aligned}\phi_{wire}^{liq} &= \phi_{liq} - \phi_{wire} \\ \phi_{wire}^{air} &= \phi_{air} - \phi_{wire}\end{aligned} \tag{2.16}$$

Once computed relative to the wire, double differencing over the relative phases ϕ_{wire}^{air}

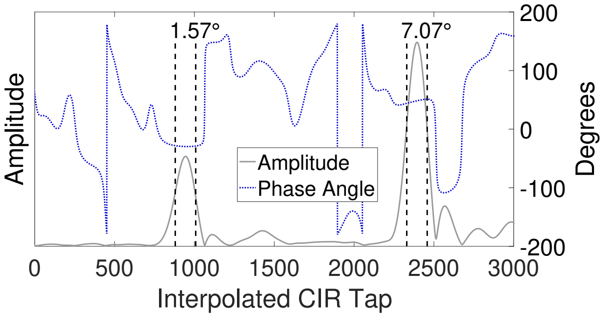


Figure 2.10: Phase is stable for many adjoining locations at the prominent peaks. An error in the estimation of the correct CIR tap results in a small phase error, but large ToF error.

and ϕ^{liq}_{wire} will provide the phase difference between the liquid and the air peaks:

$$\Delta\phi^{liq}_{air} = \phi^{liq}_{wire} - \phi^{air}_{wire} \tag{2.17}$$

Of course, the phase may wrap (called *integer ambiguity*), but we use the *relative ToF* estimate to unwrap it in the next stage.

Double Differencing RSSI: Permittivity is derived from *refractive index* (RI) and *attenuation factor* (AF). Recall that RI is essentially the slowdown of a signal in a medium, which necessitates relative ToF. However, to estimate AF, LiquID also needs to measure the signal's amplitude degradation in the liquid, w.r.t. air. Importantly, the absolute values of amplitude vary substantially due to small variations in the AGC gains at the receiver. Figure 2.11 shows this variation in the absolute received signal strength (RSS) for different packets. To handle this, we require a differencing treatment on RSSI as well, similar to ToF.

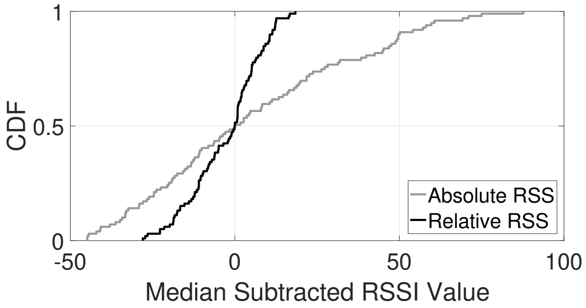


Figure 2.11: Large variation of absolute RSS, however, variation much less for relative RSS.

The reference wired path is again useful for comparison. For every packet, the amplitude of the T_{liq} sample is normalized with the amplitude of T_{wire}. Similarly amplitude for T_{air} is normalized with the corresponding T_{wire} amplitude. In essence, we scale all CIRs with the amplitude of their respective T_{wire}. The relative received signal strengths (relative RSS) can now be modeled as:

$$\begin{aligned} RSS_{wire}^{liq} &= \frac{RSS_{liq}}{RSS_{wire}} \\ RSS_{wire}^{air} &= \frac{RSS_{air}}{RSS_{wire}} \end{aligned} \tag{2.18}$$

Figure 2.11 shows how the variation in the relative RSS (across many packets) is much smaller now, while Figure 2.12 shows very good alignment of 100 (air) CIRs after scaling and aligning at sample T_{wire}.

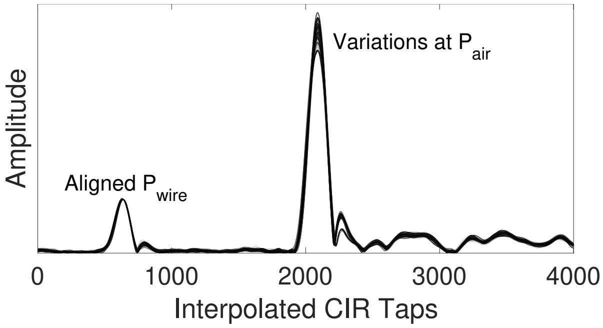


Figure 2.12: 100 overlapped "air" CIRs after scaling by the amplitude of T_{wire}. This demonstrates the opportunity to accurately compare liquid with air.

Finally, to arrive at the attenuation factor (AF), we difference a second time between the liquid's and air's RSS, giving us:

$$RSS_{air}^{liq} = \frac{RSS_{liq}}{RSS_{air}} \tag{2.19}$$

Figure 2.13 shows the result of this stage which gives the relative ToF and the relative RSS. Observe that T_{wire} serves as a precise reference against which the liquid and air can be compared, both in terms of ToF and signal strength. We can now replace the LHS of Equation 2.7 (i.e., the attenuation factor equation) with actual RSSI values obtained from our experiments.

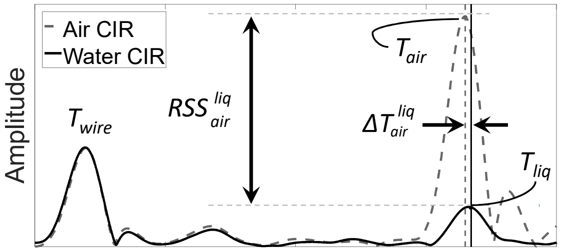


Figure 2.13: The wired reference aligns T_{wire} in the CIR, thereby enabling relative ToF and relative RSS between liquid and air.

2.4.3 Stage 3: ToF Refinement and Container Compensation

LiquID is now ready to fine tune the relative ToF (RTOF) using the relative phases, and also compensate for the effect of the plastic container on the measurements.

Relative-ToF Refinement: Since phase exhibits better precision than relative ToF (discussed earlier for Figure 2.10), we see an opportunity for refining RTOF. Observe that the RTOF is a function of the distance traveled by the signal in the liquid, which in turn equals an integer multiple of λ + measured phase ϕ. Concretely:

$$RTOF = \Delta T_{air}^{liq} = (q + \frac{\hat{\phi}_{(mod 2\pi)}}{2\pi})\frac{\lambda_0}{c} \tag{2.20}$$

where q is an integer number of wavelengths wraps, and $\hat{\phi}$ is calculated from the RTOF (i.e., $\hat{\phi} = \frac{RTOF.c}{\lambda_0}$). Also, λ_0 is the wavelength of the UWB signal in air, and c is the speed of UWB signals in air. A unique solution to this equation can be obtained using the constraints that q is an integer and $\hat{\phi}$ is bounded in $[0, 2\pi]$.

Now, we compare $\hat{\phi}$ with $\Delta\phi_{air}^{liq}$ that we had computed in stage 2 via double differencing. These should be close, but if not, we adjust q up or down to bring them closer. For instance, if $\hat{\phi} = 10°$ and $\Delta\phi_{air}^{liq} = 350°$, then we reduce q by 1. In other words, we trust $\Delta\phi_{air}^{liq}$ more and believe that the actual signal path must be $(q-1)\lambda_0 + 350°$, as opposed to $q\lambda_0 + 350°$. Thus, with this adjusted q, denoted $\hat{q}$, we refine RTOF as:

$$RTOF_{refined} = (\hat{q} + \frac{\Delta\phi_{air}^{liq}}{2\pi})\frac{\lambda_0}{c} \tag{2.21}$$

This refined RTOF is converted to velocity of UWB signals in the liquid, and averaged over 20 samples to suppress noise. As a final step, we obtain the estimated Refractive Index from Equation 2.3 as follows:

$$n = \frac{c}{v} = \frac{(\hat{q} + \frac{\Delta\phi_{air}^{liq}}{2\pi})\lambda_0}{L_{liq}} + 1 \tag{2.22}$$

Container Compensation: Of course, the container used for storing the liquid also influences the ToF, phase, and the RSS. Fortunately, we can perform all of the above measurements on an empty plastic container and obtain its complex permittivity as well. We observed a phase difference of around $45°$ and a negligible attenuation factor by introducing the empty plastic container. This results in the complex permittivity of $(3.45 - 0j)$ that closely matches typical material properties for acrylic [16]. We subtract the equivalent plastic ToF from all our measurements and modify Equation 2.10 as follows:

$$\frac{RSSI_{liq}}{RSSI_{air}} = t_E^{air\rightarrow c}.t_E^{c\rightarrow liq}.t_E^{liq\rightarrow c}.t_E^{c\rightarrow air}.\left(\frac{1}{e}\right)^{\frac{L_{liq}}{\alpha_d}} \tag{2.23}$$

With these compensations performed, we now have the values of the RSSI ratio and the expected Refractive Index of the liquid. For a different container, its permittivity, thickness,and liquid width must be taken into account and the wireless signals must be normally incident on the face of the container.

Solve for Complex Permittivity: To solve for permittivity, we rely on Equations 2.5 and 2.10). Except ϵ' and ϵ'', we know all the parameters in these equations (note that $\lambda = \lambda_0/n$, and we know the refractive index n at this point). We can therefore solve for ϵ' and ϵ'' – we employ the Matlab solver (`lsqnonlin`) and arrive at the best estimates. The next section reports on the accuracy of our estimates.

2.5 EVALUATION

2.5.1 Experimental Setup

Our experimental setup for evaluating LiquID is shown in Figure 2.1. We use the Decawave UWB Trek1000 [6] evaluation kit for our experiments. The liquid is placed inside an acrylic container located between the Tx and Rx UWB radios. The container we use in our experiments has a depth of $2\ cm$, breadth of $38\ cm$, and a height of $36\ cm$. In general, the depth should be sufficiently large to create enough measurable slow down in the signal without significantly weakening the RSSI of the signal. The length and breadth on the other hand need to be larger than the signal wavelength to avoid signal diffraction. All experiments are run in an office setting with standard furniture and multipath effects. All liquids including milk were at room temperature between 23°C and 26°C. Effervescent liquids were allowed to stand several hours in an open mouth vessel to remove all bubbles before testing. Most of the liquids were directly procured from popular grocery stores. NaCl and Glucose solutions of different concentrations, however, were prepared by mixing non-iodized salt and granulated sugar in distilled water in our lab.

Baseline: As baseline for estimating permittivity, we use a vector network analyzer. The experimental setup is shown in Figure 2.14. The setup uses an Agilent N5242A PNA-X Microwave Network Analyzer [17] which costs over $\$134000$. The PNA is connected to an HP-85070A dielectric probe [18]. The probe is dipped into the liquid which enables accurate measurement of S11 parameters. A proprietary software uses these parameters to compute the permittivity of the liquid sample. The permittivity measurements from the network analyzer are known to have a median error of 5% [5].

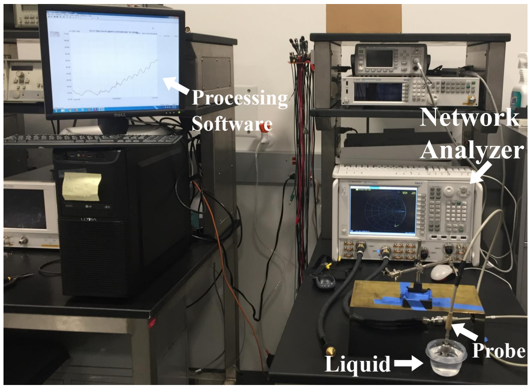


Figure 2.14: Baseline: Network Analyzer Experimental Setup

2.5.2 Results

Estimating the Permittivity

We report the permittivity estimates for 33 liquids and compare our results with the baseline network analyzer measurements. Table 2.1 shows both LiquID's and the baseline's estimates for both ϵ' and ϵ''. Figures 2.15 and 2.16 show the mismatch between LiquID and the baseline. The median mismatch in dielectric constant ϵ' is 3.29. For ϵ'' the median mismatch is 0.99. This translates into a median relative error of 9% in ϵ' and 11.9% in ϵ'', which are reasonably comparable to the baseline's error of 5%. While LiquID has slightly higher error, the LiquID setup is four orders of magnitude cheaper than the baseline setup and is non-invasive i.e. it does not require placing a probe inside the liquid.

Liquid	LiquID		Baseline	
	ϵ'	ϵ''	ϵ'	ϵ''
Distilled Water	72.34	13.16	75.64	14.7
Mineral Water	75.79	14.87	74.79	14.83
Orange Juice	55.28	18	67.38	18.77
Apple Juice	67.33	15	74.01	14.29
Grape Juice	67.06	20.43	67.16	18.74
Pepsi	71.92	17.24	74.92	20.98
Coca-Cola	67.1	16.93	71.25	18.87
Diet Pepsi	76.3	14.41	75.1	14.85
Diet Coca-Cola	79.09	14.16	80.79	19.39
NaCl 0.1M	67.75	18.63	77.68	23.96
NaCl 0.05M	65.07	15.99	75.16	18.14
Glucose 5%	54.57	14.72	71.52	15.70
Glucose 10%	65.52	15.76	68.65	16.19
Skim Milk	73.96	19.02	68.09	17.41
Milk 1%	76.22	19.34	66.87	17.12
Milk 2%	73.94	19.22	66.53	17.05
Whole Milk	70.31	19.38	64.34	16.51
Sweet Tea	77.04	16.26	71.23	16.72
Green Tea	75.26	15.65	71.54	16.28
Cranberry Juice	72.36	17.35	69.39	17.45
Coffee	68.42	12.71	73.49	12.97
Pine Sol	70.23	16.41	72.52	18.55
Isopropanol 50%	26.95	15.26	30.4	19.79
Isopropanol 70%	8.22	10.09	16.21	12.87
Isopropanol 91%	5.37	4	5.35	4.21
Ethanol 70%	15.95	12.84	17.63	15.11
Vinegar	13.88	5.46	26.84	2.17
Peanut Oil	2.22	0	2.6	0.12
Olive Oil	2.25	0	2.87	0.13
Cutting Oil	1.99	0	1.35	0
Soyabean Oil	2.25	0	2.63	0.12
Motor Oil	1.92	0	2	0
Corn Oil	2.28	0	2.57	0.14

Table 2.1: Measured permittivity with LiquID vs. Network Analyzer for various liquids.

Table 2.2 compares the complex permittivity measured by LiquID with that available in literature for some liquids [19, 20, 21, 22]. The table shows that LiquID reports permittivity measurements very close to the expected values.

Liquid	LiquID		Literature	
	ϵ'	ϵ''	ϵ'	ϵ''
Distilled Water	72.3	13.1	75.7	14.3
Ethanol 70%	15.95	12.84	16	14
Isopropanol 91%	5.37	4	5.5	4.8
Soyabean Oil	2.25	0	2.71	0.174

Table 2.2: Measured permittivity with LiquID vs. that reported in literature.

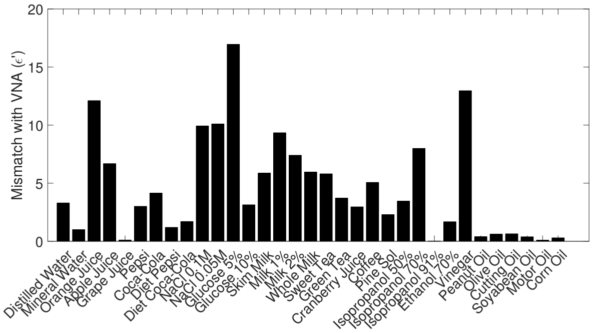


Figure 2.15: Mismatch between LiquID and baseline in ϵ'

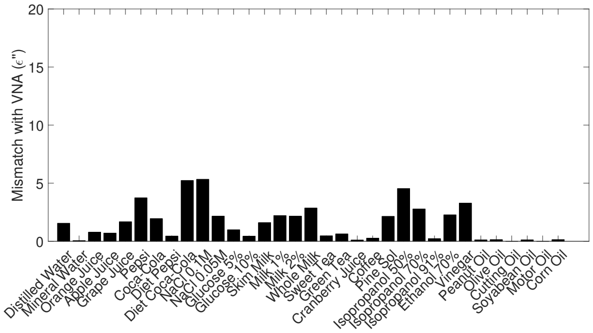


Figure 2.16: Mismatch between LiquID and baseline in ϵ''

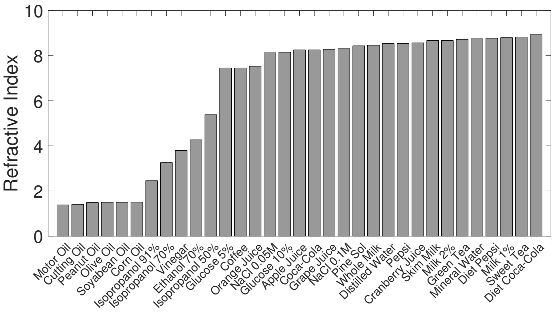


Figure 2.17: Refractive index of tested liquids

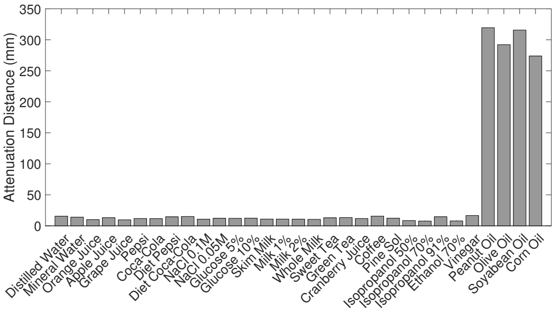


Figure 2.18: Attenuation factor of tested liquids

In order to better understand where the error comes from, we plot the refractive index and attenuation factor of the various liquids as shown in Figures 2.17 and 2.18. Recall that the permittivity coefficients ϵ' and ϵ'' are estimated from the refractive index n and attenuation distance α_d using equations 2.5, 2.7, and 2.10. However, these equations are

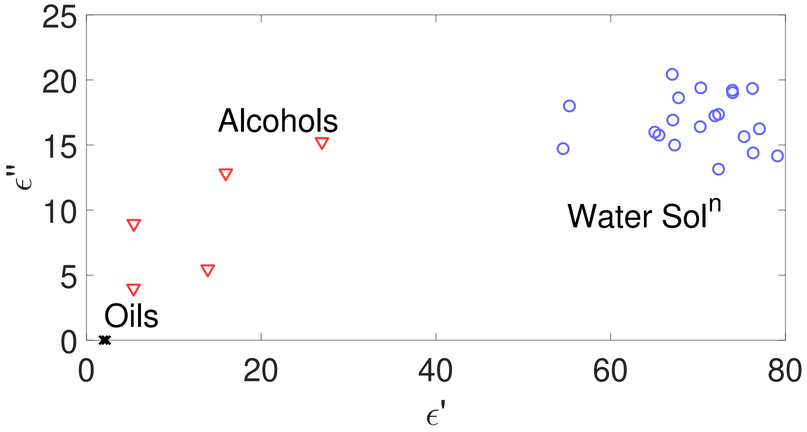


Figure 2.19: The Oils, Alcohols, and Water solutions span a large spectrum of permittivity values.

non-linear and hence any small error in measuring the refractive index or attenuation loss can lead to a large error in estimating the permittivity.

The figures also show that the liquids can be clustered into three types (Ordering of liquids in Figure 2.17 has been sorted by the refractive index to make this point clear):

- *Oils:* have a small refractive index and permittivity but a large attenuation distance and hence do not slow down or attenuate the signal much. They are *transparent* to radio frequencies.
- *Alcohols:* have a medium refractive index, attenuation distance and permittivity.
- *Water based liquids:* have a large refractive index and permittivity and small attenuation distance. They are close to *opaque* to radio frequencies.

Figure 2.19 also shows the spread of the three types of liquids over the permittivity spectrum. The figure shows that we cover a large range of permittivity values along both ϵ' and ϵ''.

Liquid Identification

Figure 2.20 shows the confusion matrix for identifying and distinguishing the 33 liquids using permittivity estimates from LiquID. Each liquid was tested 10 times and compared with the estimates from one initial experiment. Evidently, LiquID identifies most of the liquids correctly. While confusion does occur, it is limited to liquids with very close permittivity like mineral water and diet pepsi, where the discrepancy in the actual permittivity values is within the margin of measurement error. Even if we use the specialized VNA system, confusion between such liquids will persist since their permittivity values are extremely close (i.e., within the VNA's 5% error range).

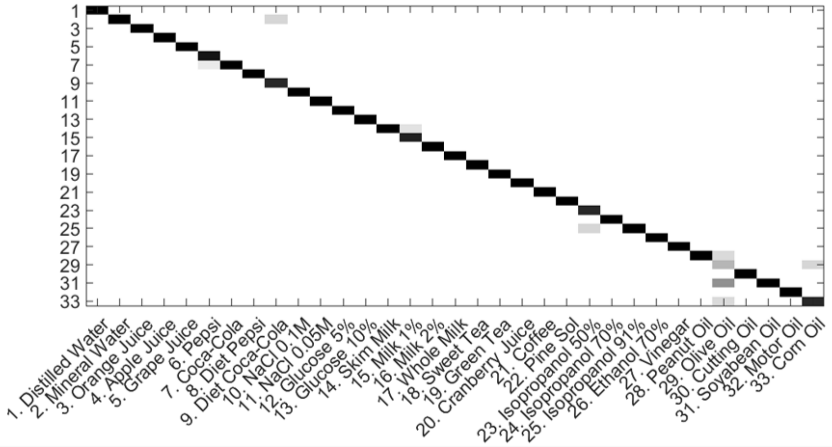


Figure 2.20: Confusion matrix for liquid identification with LiquID, using permittivity.

2.6 RELATED WORK

Optical Spectroscopy: Optical spectroscopy entails analyzing the absorption and emission properties of materials in light frequencies [23, 4]. Although very precise, the specialized equipments are bulky and expensive ($30000+), making them unsuitable for low cost operation in a ubiquitous setting. An interesting approach to liquid identification is proposed in [24] that uses photoacoustic effect. Their approach relies on shining lights of various wavelengths through liquids. However, penetration of light in dark colored liquids is limited. Instead, our approach uses radio frequency waves that can penetrate dark colored liquids including thick black oils. Moreover, [24] shows promise in a very limited set of water based solutions. In contrast, this work identifies liquids across various classes such as water solutions, oils, and alcohols. In its core, LiquID measures a fundamental physical property of all substances and does not just classify liquids.

Impedance spectroscopy at RF: These techniques are based on impedance responses of a material to an applied RF field (example techniques include co-axial probe methods, free space methods, transmission line and reflection methods, resonant techniques [25, 26, 27, 28, 29]). A popular approach uses a co-axial probe dipped into the liquid while a vector network analyzer (VNA), connected to the probe, measures permittivity [5]. Impedance mismatch at the probe-material boundary causes reflections whose properties determine the liquid's permittivity. While useful for studying permittivity across a broad RF spectrum, the equipment is not only expensive and bulky, but also invasive. In contrast LiquID provides a non-invasive and low cost alternative.

UWB and RFID Approaches: Prior works have used RFID and UWB wireless signals to identify material properties such as oil adulteration[7], attenuation due to building materials[8, 30] (wood, gypsum, glass), and salt concentration[9]. Closest to our work is Tagscan [10], which uses RFID signals to classify 10 liquids. Tagscan's approach can only classify liquids whose phases differ by certain relationships. Specifically, two liquids that cause different number of 2π phase wraps to the signal, cannot be discriminated. Thus, the classical problem of *integer ambiguity* still remains, allowing TagScan to discriminate a few selected liquids. In contrast, LiquID exploits high resolution ToF of UWB to resolve the integer ambiguities, and in combination with precise double-differencing methods, estimates the actual complex permittivity ($\epsilon' + \epsilon''$). The technique scales to any liquid (33 liquids reported in this paper), while requiring no training for classification – the estimated permittivity can be directly compared against ground truth.

ToF, Phase, and RSSI: A number of localization, and gesture tracking papers leverage ToF, phase, and RSSI measurements not only in RF [31, 32, 33, 34, 35], but also in acoustics [36, 37, 38]. SAIL [31] uses ToF measurements for indoor localization. WiTrack [34] detects human motion based on ToF computed from FMCW Radar[34]. SafetyNet [39], and [33] uses phase for computing orientation of drones and objects. In contrast, the ToF, phase, and RSSI accuracy requirements for LiquID is very high, in the granularities of few mm and 0.3 dB. LiquID uses a multi stage signal processing pipeline to achieve its requirements.

2.7 POINTS OF DISCUSSION

Closing the Gap in Permittivity Error.

Permittivity error is sourced in ToF, phase, and RSSI errors. While improvements may be possible via higher sampling frequency (i.e. bandwidth) or longer point Fourier transforms, we focus on the room for algorithmic improvement. Specifically, ToF and phase errors can be improved by further optimizing the choice of the CIR sample at which the signal arrived. Of course, this is a function of multipath and diffraction. While we partially compensate for multipath, diffraction is harder since diffracted paths could arrive at the same time (or even earlier) than the slowed-down path through the liquid. This also affects RSSI errors, especially because the path through the liquid is already weak. Isolating diffraction is hard; we leave this to future work.

Can Any Sized Container be Used?

The dimensions of our container are chosen carefully so that signals do not impinge obliquely on the vertical surface, while also ensuring that RSSI and slow-down are reasonable. Arbitrary shaped containers are extremely challenging (according to [11], "the general case of arbitrary dielectrics and oblique incidence is of great complexity"). Moreover, when the container is below a threshold size, the diffracted signals may completely drown the through-liquid signal. LiquID for arbitrary containers remains an open problem.

Permittivity versus Frequency.
Signal slowdown, and hence permittivity, is a function of frequency. For our high bandwidth signals, the measured permittivity is actually an aggregate of many frequency-specific permittivities. With narrow band signals, the permittivity estimates can be improved at the expense of loosing ToF precision (recall high bandwidth is necessary to precisely estimate ToF). This is a tradeoff and perhaps new techniques are feasible that leverage the best of both worlds. We leave this to future work.

Beyond Liquids.
Our core techniques should generalize to solids, with suitable modifications to the physics model and parameters. In fact, the signal slowdown may be more pronounced, aiding in more precise ToF calculation. The only technical issue may be around RSSI – solids may attenuate RSSI more than liquids, requiring a higher-power signal, or narrower material width. A follow up work needs to thoroughly address the case for solids.

Considerations for a Mobile Form Factor.
Fitting LiquID into a mobile form factor entails incorporating the space, energy, and computation constraints of such platforms. An UWB chip occupies a $4mm \times 4mm$ area and weighs $0.105gm$ [40]. Whereas we have used half-wavelength dipole antennas in our experiments, much smaller antennas are available [41]. UWB is a low-power protocol [42] and the decawave chip is rated to consume a maximum of $120mAh$ [40]. The signal processing blocks required by LiquID are already present on mobile devices [43]. Pipelining the signal processing with fetching of the CIR data can allow LiquID to run at near-realtime. Finally, we envision a mobile device with an antenna connected to an extensible wire. A liquid container is placed between the mobile device and this antenna to identify the liquid.

2.8 CONCLUSION

This work shows the feasibility of identifying liquids by analyzing UWB signals passing through it. We measure the time of flight of the signal and combine with its phase and RSSI to ultimately model the permittivity of the liquid. Given permittivity serves as a signature, it is now possible to identify the liquid without inserting probes into it. Our solution is realtime (sub-second latency), cheap ($\approx \$150$), and lightweight (few pounds), underpinning a variety of applications. Our next step is focussed on analyzing more complex liquid mixtures, such as impure drinking water, blood, saliva, and other biologically relevant liquids.

CHAPTER 3: LOCALIZING USING WIRELESS SIGNALS: A PRIMER

The earth's surface is logically divided into a grid defined by latitudes and longitudes. In the absolute sense, localization is the process of placing an object on this geographic coordinate system. Historically, this has been performed using surveys starting from references and benchmarks—carefully erected structures that denote a known geographic point. More recently, almost all localization is performed by GPS satellites. Interestingly, a lot of the principles of localization have actually stayed the same over centuries. In this chapter, we will discuss this basic principle followed by challenges in localization using wireless signals.

3.1 TRILATERATION BASICS

An object's location can be determined by measuring its distance from three non-collinear reference positions. This idea was used historically when using geographic benchmarks and the same idea is still applicable when using GPS. Starting from each reference position, or anchor, the distance measurement, or range, can be represented as a circle around that anchor with radius equal to the measured distance. The point of intersection of these three circles then represents the object's location. Figure 3.1 shows this basic notion of localization where an object, T, is localized from three anchors, A_1, A_2, and A_3.

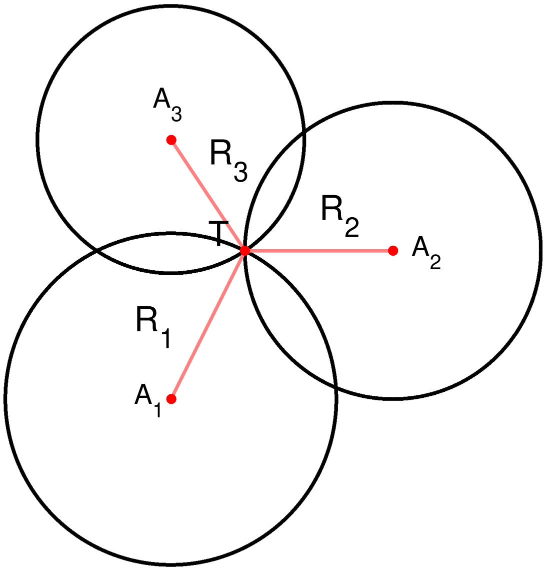


Figure 3.1: The basic principle of localization using trilateration

Mathematically, the location of T (T_x, T_y) is given by simultaneously solving the following non-linear equations:

$$R_1 = \sqrt{(A_{1_x} - T_x)^2 + (A_{1_y} - T_y)^2} \tag{3.1}$$

$$R_2 = \sqrt{(A_{2_x} - T_x)^2 + (A_{2_y} - T_y)^2} \tag{3.2}$$

$$R_3 = \sqrt{(A_{3_x} - T_x)^2 + (A_{3_y} - T_y)^2} \tag{3.3}$$

Practically, ranging measurements are seldom exact, resulting in no unique intersection point. Hence, it is preferable to obtain an estimate of the object's location using a minimization error function such as:

$$f = \underset{(T_x,\ T_y)}{\operatorname{argmin}} \sum_{i=1}^{3} \left(\sqrt{(A_{i_x} - T_x)^2 + (A_{i_y} - T_y)^2} - R_i \right)^2 \tag{3.4}$$

While this basic principle holds in general and provides us the best estimates of the object's location, we will see later that inexact ranging measurements can have a significant impact on the localization accuracy depending on the anchor geometry.

3.2 WIRELESS DISTANCE MEASUREMENTS

We assumed above that the range between the object and anchors is given to us. How do we obtain this range? We intend to use propagation delay of wireless signals between two devices to obtain this distance. There are three challenges in measuring the propagation delay: (a) since electromagnetic waves travel at the speed of light, we need extremely fast clocks to measure propagation delays at centimeter level accuracy, (b) since clocks at the two devices are unsynchronized, we need a mechanism to compensate for clock offsets, and, (c) since clocks can drift at unpredict able rates, we need a mechanism to compensate for clock drifts. Whereas the first challenge is solved using nanoseconds clocks, the other two challenges necessitate a three-message exchange. Lets understand why.

A single wireless message sent between two devices can measure time-of-flight only when their clocks are synchronized. Without clock synchronization, the offset between the two clocks is indistinguishable from the propagation delay. A ping-pong involving two messages, called POLL and RESP, as shown in Figure 3.2(a) can overcome this issue, but will have ranging error proportional to the turn-around time. Adding one more message, called FINAL, helps reduce this error.

Fig. 3.2(b) shows the state-of-the-art three-message protocol, called symmetric two way

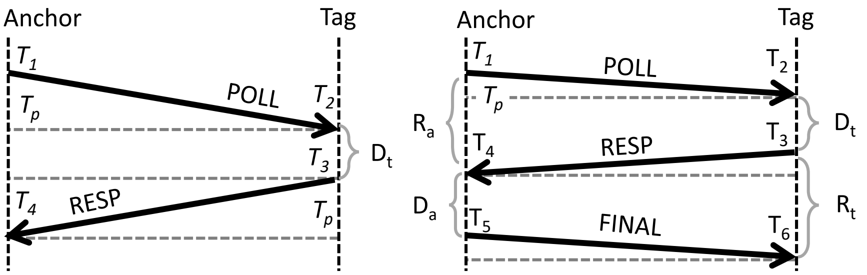


Figure 3.2: (a) A simple 2-message exchange protocol (b) The two-way ranging (TWR) protocol.

ranging (TWR) [44, 42], that mitigates the drift issue. The initiator, called the anchor, sends a POLL message at time T_1 that is received by a responder, called tag, at time T_2. The tag requires a finite turn-around time, D_t, to send a RESP back to the anchor at time T_3. The anchor receives RESP at T_4, and sends a FINAL message. All packets are time-stamped and contain the reception time of the previous message. The ranging time, time between sending a message and receiving a reply, is denoted by R_x, while, the turn-around delay is denoted by D_x. The propagation delay T_p is given by:

$$T_p = \frac{((R_a - D_t) + (R_t - D_a))}{4} \tag{3.5}$$

It can be shown that ranging error depends on the product of the *difference* in the turn around times and the difference in the clock drifts (δ_a, δ_t) [44]:

$$Error \propto (D_a - D_t)(\delta_a - \delta_t) \tag{3.6}$$

Best accuracy is achieved when the two turn-around times are almost equal. Further improvement is possible, as shown in [45], and the ranging error can be made independent of the turn-around times using:

$$T_p = \frac{(R_a * R_t - D_a * D_t)}{R_a + R_t + D_a + D_t} \tag{3.7}$$

$$Error \propto (\delta_a - \delta_t) \tag{3.8}$$

This modification permits unequal turn-around times without affecting ranging accuracy. Once the propagation time is calculated, the distance is simply propagation time multiplied by the speed of light. We will modify this TWR protocol to suit our needs when localizing a group of devices together.

3.3 HARDWARE FOR ACCURATE MEASUREMENTS

Two-way ranging protocol described above works only if it is possible to exactly send out wireless packets at predetermined time instances. Further, its accuracy depends on exact reporting of the receive time. Both these properties require the use of high precision clocks that are present in ultra-wideband (UWB) radios due to very high sampling requirements. In this work, we use the Decawave Ultra-wideband TREK1000 evaluation kits [6] at 4GHz central frequency and 1GHz bandwidth. These devices are pre-programmed to perform the TWR protocol and report distances between two devices. We completely redesigned the ranging protocols for various aspects of this work according to our requirements. We will discuss the specific modifications in subsequent chapters. The UWB platform is shown in Figure 3.3(a).

One key benefit of using Decawave evaluation kits is that we can also obtain the channel impulse response (CIR) computed by the device. It is represented as a set of complex numbers that together represent the entire wireless channel in the vicinity; the reflections from various objects in the environment are all captured in this CIR as shown in Figure 3.3(b). In localization, we will be most interested only in the shortest path between the devices. However, we will see the use of other paths of the CIR when employing wireless signals for material identification.

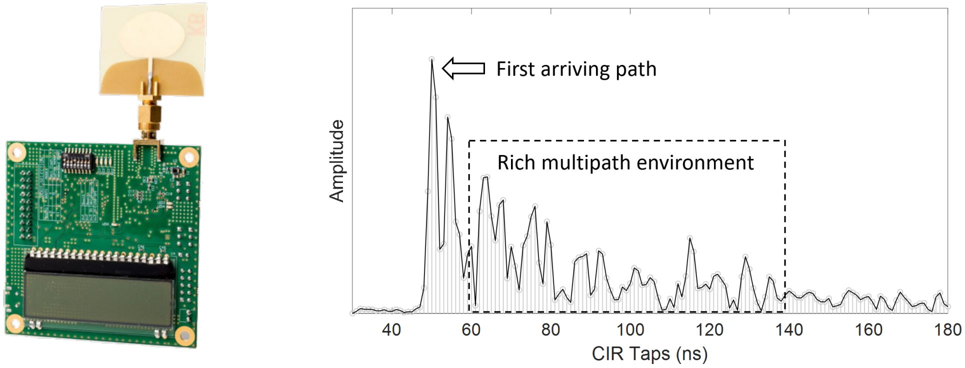


Figure 3.3: (a) The Decawave Trek1000 UWB platform (b) An example CIR of an office environment

CHAPTER 4: TRACKING BALLS AND PLAYERS IN A SPORTS ARENA

Sports analytics is a thriving industry in which motion patterns of balls, racquets, and players are being analyzed for coaching, strategic insights, and predictions. The data for such analytics are sourced from expensive high-quality cameras installed in stadiums, processed at powerful backend servers and clouds. We explore the possibility of significantly lowering this cost barrier by embedding cheap inertial and magnetic sensors (IMU) and ultrawide band (UWB) radios inside balls and players' shoes. If successful, real-time analytics should be possible anytime, anywhere. Aspiring players in local clubs could read out their own performance from their smartphone screens; school coaches could offer quantifiable feedback.

4.1 TRACKING A CRICKET BALL

In approaching this problem top-down, we develop multiple wireless and sensing modules, and engineer them into a unified solution. The technical core of our system relies on using ultrawide band (UWB) radios to compute the time of flight (ToF) and angle of arrival (AoA) of the signals from the ball. When this proves inadequate, we model the ball's physical motion as additional constraints to the underdetermined system of equations. Finally, we fuse all these sources of information into a non-linear error minimization framework and extract out the parameters of ball trajectory.

Fig. 4.1 illustrates the overall deployment in real settings. UWB signals are exchanged between the ball and anchors to compute the balls range as well as the angle of arrival (AoA) from the phase differences at different antennas of a multiple-input multiple-output (MIMO) anchor. The range and AoA information are combined for trajectory estimation. For spin analytics, the sensors inside the ball send out the data for off-ball processing. Players in the field can optionally wear the same IMU/UWB device (as in the ball) for 2D localization and tracking.

4.1.1 System Design: 3D Trajectory Tracking

Location related analytics are also of interest in Cricket. 3 main metrics are: (1) distance to first bounce, called *length*, (2) direction of ball motion, called *line,* and (3) *speed* of the ball at the end of the flight. These metrics are all derivatives of the ball's *3D trajectory*. Our approach to estimating 3D trajectory relies on formulating a parametric model of

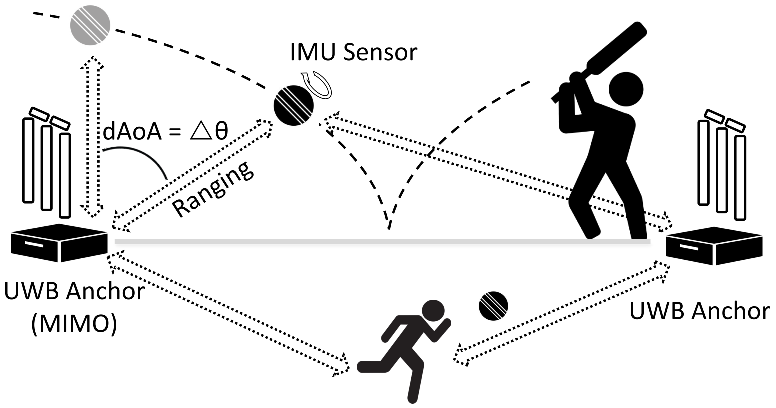


Figure 4.1: Two anchors and a ball deployed on the ground, while players optionally have the device in their shoes.

the trajectory, as a fusion of the *time of flight* (ToF) of UWB signals, *angle of arrival* (AoA), physics motion models, and DoP constraints (explained later). A gradient decent approach minimizes a non-linear error function, resulting in an estimate of the trajectory. We present technical details next.

Ranging with UWB

The Decawave UWB radios offer time resolution at $15.65ps$. With modest engineering, we were able to compute the ToF and translate it to range measurements (with $15cm$ error). Briefly, the ball sends a POLL, the anchor sends back a RESPONSE, and the ball responds with a FINAL packet. Using the two round trip times, and the corresponding turn-around delays, the time of flight is computed without requiring time synchronization between the devices. Multiplied by the speed of light, this yields the ball's range. We improved the ranging update rate to about $150Hz$ by actively ranging with a single MIMO anchor and the other anchor passively listening to the message exchange between the ball and the MIMO anchor.

Observe that UWB ranging is available from only 2 anchors (placed at the two wickets) and therefore inadequate to resolve the 3D location of the ball. Additional anchors cannot be placed on the ground since it will interfere with the motion of the ball and fielders, while placing anchors outside the field ($90m$ away from the wickets) significantly degrades SNR and ranging accuracy. Figure 4.2 shows the intersections of the 2 anchor measurements, i.e., circles formed by the intersection of two spheres centered at the anchors. At a given time, the ball can lie on any point of a circle. Given that the initial position and velocity of the ball is unknown, many 3D trajectories can satisfy these constraints.

Exploiting Angle of Arrival (AoA)

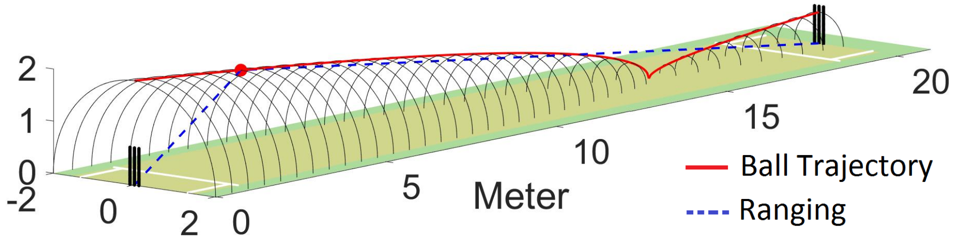


Figure 4.2: Intersections of ranging measurements leave one location dimension unresolved.

The MIMO antennas at the anchors are capable of synchronized phase measurements of the incoming signal. Figure 4.3 shows how the phase difference ϕ is a function of the difference in signal path (p_1 and p_2), which is in turn related to AoA, θ. Thus, we have:

$$d\cos(\theta)\frac{2\pi}{\lambda} = \phi \tag{4.1}$$

$$\cos(AoA) = \cos(\theta) = \frac{\phi\lambda}{2\pi d} \tag{4.2}$$

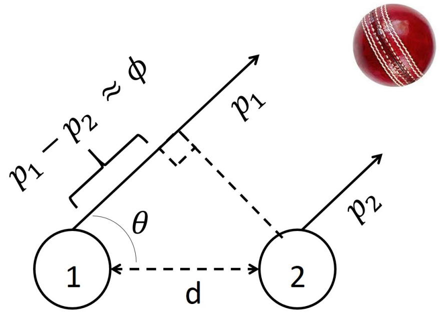


Figure 4.3: AoA can be measured from phase differences.

We employ a MIMO receiver only on the bowler side (the other anchor cannot be utilized since it gets significantly interfered by the batsman, corrupting phase measurements). Thus, we refine the previous ambiguous estimates of the trajectory by including the angle of arrival.

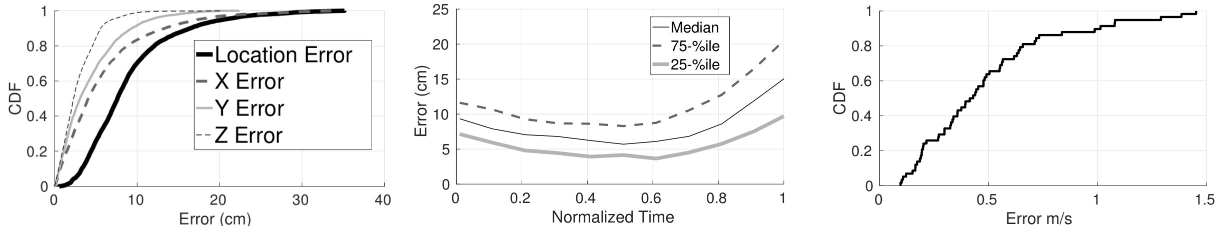


Figure 4.4: (a) CDF of location tracking errors. (b) location error degrades slightly toward the end of ball flight. (c) CDF of ball speed error.

Figure 4.4(a) quantifies the location error (LOE) across 50 different throws – the median error is $8cm$. We also report the errors on each of the directions: Y in the direction of the throw, Z being vertically upwards, and X is perpendicular to Y and Z. The median X, Y, and Z axes errors are $4.5cm$, $3.4cm$ and $2.39cm$ respectively. The X axis errors are maximum due to DoP effects, however, AoA lowers it to a reasonable value.

4.2 INFRASTRUCTURE FREE PLAYER TRACKING

We used a pair of fixed anchors equipped with angle of arrival to localize the cricket ball. However, fixed anchors may not always be available, for example on a beach or on an informal sports field. Can a team still localize its members and determine the group's formation? For instance, a team of soccer or basketball players may seek their precise configurations during a game, valuable to coaching and sports analytics applications. Outdoors, GPS may not be adequately precise (typical accuracy is a few meters). Indoors, e.g., in a basketball court, GPS may simply not be available. This work seeks to develop a mobile P2P localization system, where the mobile entities cooperatively estimate their team topology. Our wearable devices (i.e., Ultra-wideband (UWB) radios worn by team members) exchange wireless messages to compute pairwise distances until adequate distances are available to "pin down" the topology. Fig. 4.5 illustrates an example topology that form the output of our system.

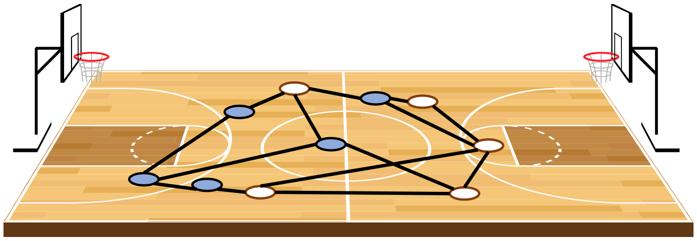


Figure 4.5: We envision real-time visualization of group topologies such as in a basketball game.

There are a number of challenges in realizing this system.

1. In trilateration, a node is deemed to be at the intersection of three *circles* formed by range measurements from three anchors. However, in practice, ranging imperfections smudge the circles into *annular rings* and the intersection point into a confusion region (see Fig. 4.6). The choice of anchors affects the area of this region, impacting localization accuracy, in a phenomenon called dilution of precision (DOP). When using static infrastructure, careful placement of anchors can minimize DOP—a one time decision. However, in P2PLoc where all nodes including the anchors are constantly

moving, the anchor-set must change dynamically based on the DOP susceptibility of various anchor combinations in the current topology.

2. Body occlusions degrade link quality to the point where a TWR attempt can fail (wasting time and resources). We therefore need a mechanism to estimate link quality between all pairs of nodes to select a good anchor set, in addition to a DOP based selection criteria.

3. P2PLoc's update rate, the number of location estimates per second, depends on the time consumed in the ranging process. Whereas every node must range with at least 3 other nodes, TWR is a 1:1 ranging protocol; we modify it to range a group of nodes.

4.2.1 Core Challenges and Solutions

Trilateration-based localization techniques comprise of measuring a node's distance (or range) from at least three reference locations (often called anchors). In our case with mobile nodes, localization accuracy is impacted by the choice of anchors, mainly due to DOP and body occlusions. A set of anchors that are least affected by DOP or occlusion currently may soon become ineffective, warranting new anchor selection. We discuss these issues next, followed by discussion on a third issue around refresh rate, i.e., the need to frequently localize the topology to keep up with fast-moving nodes.

Dilution of Precision (DOP)

Recall that a device (tag) can be unambiguously localized by measuring its distance from three non-collinear locations (anchors) by trilateration. However, practical ranging measurements are not perfect and have a small degree of error ΔR due to hardware noise and other imperfections. The ideal picture of three intersecting circles now has to be modified to three intersecting annular rings as shown in Fig. 4.6(b). This error acts as a distortion factor around the true range and results in an error in the location estimate around the true location of the tag. The central dark colored region in Fig. 4.6(b) denotes the common intersection of the three annular rings, the region of confusion, introduced by a ΔR error in each ranging measurement.

The shape and area of this region can vary dramatically based on the relative location of the anchors and the tag, causing a large variation in the final localization error. For example, observe the shape of the dark region in the four anchor geometries shown in Fig. 4.7. The average localization error in the first example is almost 10% of that in final example.

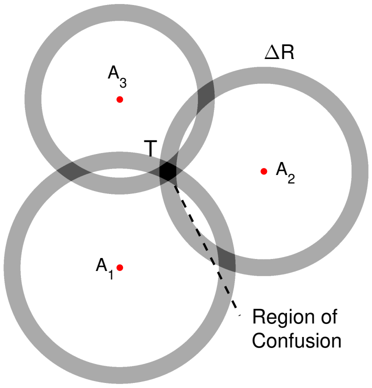


Figure 4.6: Ranging errors introduce a region of confusion

This concept of variations in the localization error depending on the geometric configuration of the nodes is known as *dilution of precision* (DOP). To improve localization accuracy, P2PLoc must choose anchor nodes that will minimize DOP for the whole topology. Further, since all nodes are in constant motion, this choice must be updated periodically.

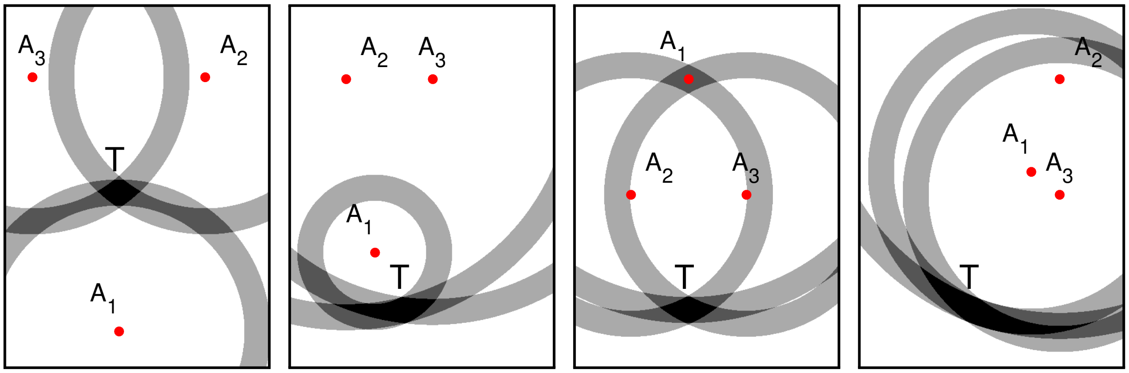


Figure 4.7: Localization error depends on anchor geometry.

DOP is commonly discussed in the context of GPS [46] where individual mobile devices decide the set of satellites to range with. However, GPS-DOP algorithms assume that satellites are extremely far away from the device and therefore rely only on the angles formed between a device and the satellites. In our case, where anchors are much closer, configurations with the same angles can have very different geometries (see Fig. 4.8) affecting the confusion region despite identical GPS-DOP. Therefore, instead of using GPS DOP algorithms, we will periodically calculate expected localization error through Monte Carlo simulations.

Anchor choice depends on the overall expected localization error due to DOP, which we can periodically calculate based on the latest known locations of each node. The previous topology gives a set of all-to-all distances. We then perform a Monte Carlo simulation using these distances. It calculates the expected localization error for every possible combination

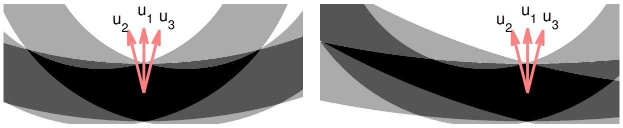


Figure 4.8: Two anchor configurations with the same GPS DOP metric but different regions of confusion.

of anchor-set by perturbing these anchor-tag distances with a Gaussian error. The anchor combination that has the least median localization error for the entire topology is then selected for the next round.

We have observed about $3\times$ improvement in localization accuracy with anchors chosen after DOP considerations, compared to randomly choosing the anchors. Clearly, much work remains in improving the efficiency of this short-distance DOP algorithm, however the gains are promising. We alleviate some computational burden by solving DOP once every two seconds, which is acceptable since DOP changes slowly.

Occlusions

Human bodies block wireless signals (i.e., signals get absorbed and they also slow down through the liquids inside the body), affecting the quality of ranging. Fig. 4.9 shows measurements where ranging accuracy for nodes behind a human body is significantly worse, compared to other nodes that are not occluded. This suggests that with P2PLoc, anchors that can experience line of sight (LOS) to more number of tags are desirable. Unfortunately, determining the anchors with least occlusion also requires knowledge of the team's topology.

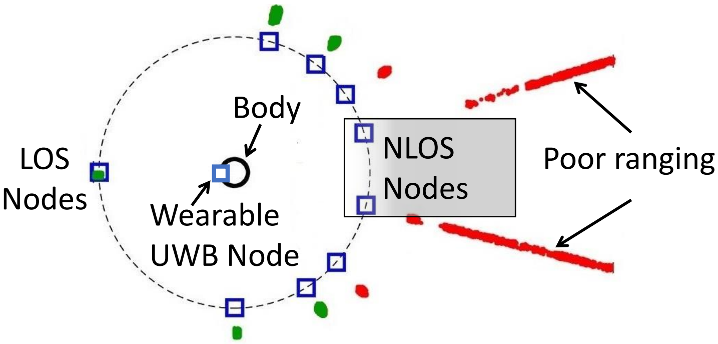


Figure 4.9: A person's body occludes a cone of about 30°.

As seen in Fig. 4.9, ranging estimates can be erroneous for occluded links since the node might measure some non-line-of-sight paths (NLOS). We wish to avoid using device-pairs with NLOS links to limit its impact on the overall localization accuracy. Occlusion avoidance depends on our ability to classify links as LOS or NLOS from link-quality information. Typically, this is available only for links on which a message exchange has occurred. Therefore, without all-to-all message exchanges in P2PLoc, the link-quality information may remain incomplete. Fortunately, UWB devices can overhear all the ongoing communication on the channel and therefore, estimate link-quality even in absence of an actual message exchange on that link. Since all nodes in the network transmit at least one packet, every node obtains link-quality metric with all other nodes. Each node then uses the obtained channel impulse response (CIR) to decide whether each link is LOS or NLOS [47], producing an occlusions list. We then choose anchors that minimize use of occluded links.

Frequent Ranging Operations

Each round of topology-localization should ideally be fast so that P2PLoc can keep up with fast moving nodes (otherwise, the topology estimate would become erroneous due to use of stale data). Recall the core ranging operation form Chapter 3 and note that it is a 1:1 distance measurement operation between a node and an anchor. In light of the TWR protocol, a single distance measurement essentially requires 3 messages. For a group of nodes to measure all-to-all distances, the total number of messages exchanged, and therefore the time consumed, is obviously non-trivial. The overall system update rate can be improved by reducing this message exchange.

For localizing a team of n nodes, each node needs to make 3 ranging measurements to 3 other nodes. This leads to $3n$ ranging measurements. However, when A ranges to B, since B also can compute its range from A, we only require $\lceil \frac{3n}{2} \rceil$ measurements. Thus, given TWR's 3-message exchange (POLL, RESP, FINAL), a total of $\lceil \frac{9n}{2} \rceil$ messages will be required for n-node topology. This can considerably slow down P2PLoc, hence, it is imperative to reduce the number of messages while still obtaining enough distance measurements.

The improvement opportunity arises from the broadcast nature of wireless packets. Since all POLL packets used by a particular node are identical—there is no tag specific information—we can collapse multiple POLLs into one single broadcast. In other words, an anchor will broadcast a single POLL that will be received by all the tags simultaneously, which can then take turns to respond, avoiding collisions. We delay the FINAL message until all RESPs are received and send all receive timestamps in one coalesced FINAL message. Such a protocol modification is feasible since, from Eq. 3.7, *localization error is independent of turn-around times.* This enables ranging with $(n-1)$ nodes using just $2+(n-1)$ messages instead of the $3(n-1)$ messages required in the TWR scheme. One such *cascade* results in all tags obtaining their distance from one anchor.

Each tag needs distance measurements from at least 3 anchors. If 3 anchors perform this cascade sequentially, we will still require $3(2+(n-1)) = 6+3(n-1)$ messages. Can this be further optimized? Observe that, instead of a tag sending separate RESP messages to each anchor, all these can be compressed to a single RESP message if the anchors *pipeline* their POLL and FINAL messages. This reduces the total number of messages exchanged to $3 + (n-1) + 3$. We call this a pipelined cascaded TWR scheme and is shown in Fig. 4.10.

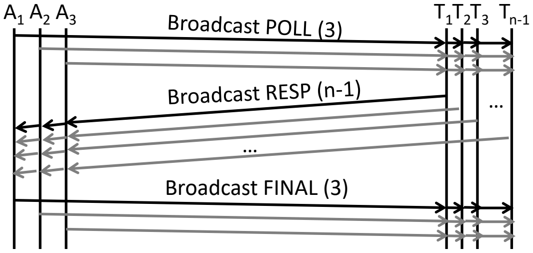
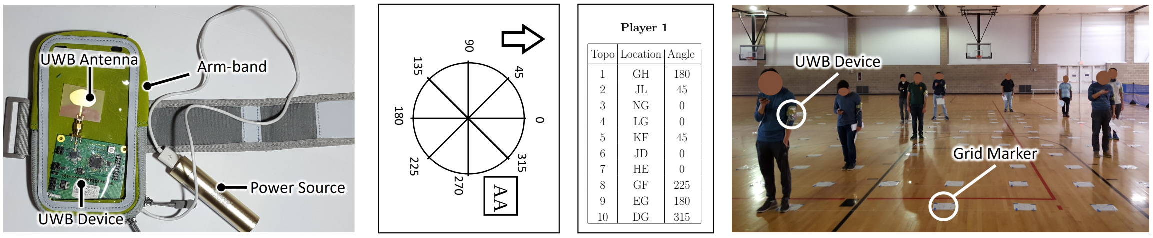


Figure 4.10: Optimized ranging for n-node topology.

Note that the optimal 3-edge cover that results in $\lceil \frac{3n}{2} \rceil$ measurements would involve edges originating from various nodes requiring more than 3 anchors in general. In contrast, cascading and pipelining is most effective when only 3 nodes behave as anchors. Thus, while P2PLoc requires twice the number of optimal measurements necessary, the number of *messages exchanged* is optimal.

Of course, the FINAL message in P2PLoc is longer than in TWR scheme since it now contains $(n-1)$ receive timestamps instead of just one. This increased transmission time, however, is a small price to pay compared to the overhead of separate messages (with preamble and headers).

4.2.2 Implementation Details



Player 1

Topo	Location	Angle
1	GH	180
2	JL	45
3	NG	0
4	LG	0
5	KF	45
6	JD	0
7	HE	0
8	GF	225
9	EG	180
10	DG	315

Figure 4.11: (a) The UWB device and its wearable casing. (b) Grid marker sheet (c) Each participant's unique list of per topology locations. (d) Volunteers following the topology instructions.

We have implemented the pipelined cascaded TWR on 11 Decawave Trek1000 UWB devices [48], each powered by a 3200mAh battery. One of these is connected to a Lenovo laptop and is together designated as the leader UWB device. The laptop runs the P2PLoc algorithm in Matlab R2016a, and chooses the anchors for the next ranging round. The UWB devices complete ≈ 7 rounds of measurements per second for visualizations.

To ground our solution in one specific application, we have chosen player localization in sports. Each UWB device is encased in an arm-band worn by the players. Fig. 4.11(a) shows one such node along with its casing and battery. To obtain precise ground truth without expensive infrastructure such as Vicon [49], we emulate mobility step-by-step through multiple static topologies. A $15m \times 15m$ grid is plotted on a basketball court with a marker at each grid location (Fig. 4.11(b)). We invited 10 volunteers to play a game on this arena (Fig. 4.11(d)). Each participant was given instructions about their position and orientation for each topology (Fig. 4.11(c)). The game involves each participant standing on their designated marker and all players together moving to the next listed location, creating 21 distinct topologies. Whereas P2PLoc already allows fast mobility, obtaining accurate time synchronized ground truth for such motion is challenging.

4.2.3 Evaluation: Localization Error

We evaluate P2PLoc with and without human occlusions. First, 10 volunteers wore UWB devices providing significant occlusions. Second, the UWB devices were mounted on tripods creating occlusion-free topologies. Procrustes analysis without scaling was then used to best fit the orientation and translation of the ground truth to obtain absolute location error [50]. Without occlusions, the median localization error is ≈ 15cm, while it is ≈ 35cm with occlusions. Fig. 4.12 shows the average error for each of the 21 topologies. Except for 1 topology where some nodes were completely occluded, all 10 nodes were successfully localized in other topologies.

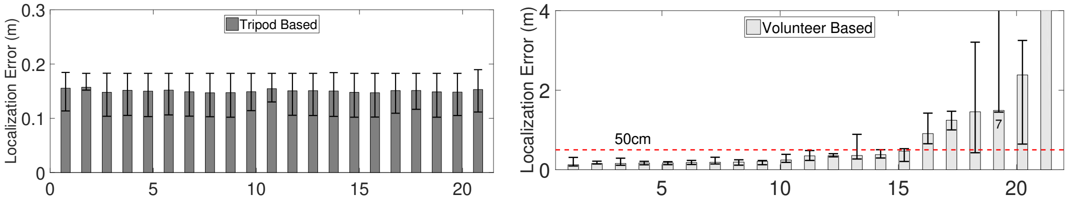


Figure 4.12: Localization error for all topologies in occlusions-free and occlusions-prone experiments.

4.3 CONCLUSION

A sports arena offers a challenging environment for wireless localization. In general, sports demand a real-time localization solution which cannot rely on historic data for its results. Further, there are not many places inside the arena to deploy anchor nodes. We have shown how the need for three fixed anchors for localization can be relaxed, and instead, AoA available from a MIMO anchor can be leveraged on a cricket field where placing two anchors is possible. In other sports, each of the players can themselves function as mobile anchors and yet enable reasonable tracking of all the players. We believe this groundwork will enable ubiquitous sports tracking systems in the future, improving the game for the players, viewing experience for the audience, and effectiveness of coaching. The primitives of fast ranging and peer-to-peer infrastructure free localization developed in this work can have impact beyond the sports field; these ideas can be adapted for tracking of soldiers on a battlefield, or tracking autonomous robots on treacherous terrains.

CHAPTER 5: TRACKING EMERGENCY RESPONDERS INDOORS

5.1 INTRODUCTION

First responders are integral to the safety and security of any community and to the society at large. However, they often find themselves in precarious and unknown environments, which poses a threat to their own safety (e.g. "entrapment" [51] faced by fire-fighters). Being able to accurately track first responders in indoor environments, allows a commander outside to better visualize and direct his responders appropriately. This not only helps address the situation efficiently but also ensures safety of the responders themselves—the latter can now view and track their own location with respect to the rest of the team.

The topic of indoor localization has seen many solutions in the past decade [52, 53, 54]. These can be broadly categorized under those that rely on indoor infrastructure (e.g. multitude of access points (APs), radio waves/acoustic/infrared beacons, etc.) and those that do not (e.g. leveraging cellular base stations (BSs), GPS satellites, inertial sensors, etc.). While the latter can be applied to our target environment, they either offer less-than-desirable accuracies (e.g. tens of meters with cellular BSs), or are not functional indoors (e.g. GPS). Inertial sensors carried by responders are a possibility, but suffer from poor accuracy as well ($\approx 10-50\,$m, due to drift over time), without periodic recalibration and resetting to known indoor reference points. Further, the lack of access to multiple stationary APs/BSs, prevents these solutions from accurately tracking *mobile* responders in real-time. Hence, notwithstanding the plethora of prior solutions, our target environment requires a new, robust, (indoor) *infrastructure-free* solution that can *accurately* ($\approx$1–2 m) track *mobile* responders in unknown, indoor environments from outside.

This motivates us to design a localization-tracking system from scratch, paving the way for two key design choices: (i) modality of localization, and (ii) wireless technology for localization. The lack of indoor infrastructure support, and the need to quickly deploy and localize responders in 3D, across multiple floors of a building (from outside), makes unmanned aerial vehicle (UAV), an ideal platform for the task. The UAV can serve as a virtual mobile infrastructure that is deployed on-demand, outside the building to localize the responders inside. For the choice of wireless technology, we summarize their pros-cons in Table 5.1. We refer to only techniques that allow long distance localization that would be applicable in our application. While lower frequencies (e.g. LTE) offer better

	LTE[55]	WiFi[56]	UWB[57]	mmW[58]
Accu	> 20m	5m	10cm	1cm
Range	> 1km	100m	50m	40m

Table 5.1: UWB offers the best tradeoff between accuracy and penetrability among all RF localization technologies.

indoor penetration/coverage (e.g. 1 Km) from outside, they are limited by available bandwidths (tens of MHz) and hence accuracy (tens of meters [55]). Higher frequencies (e.g. mmWave, > 20 GHz) offer high accuracies (tens of cm) from large (GHz) bandwidths, but suffer from high attenuation (does not work with blockages, not accounted for in Table 1). Ultra-wideband (UWB) operates in $3 - 10$ GHz and offers a 1 GHz bandwidth, thereby striking a good balance between accuracy (tens of cm) and indoor penetration (tens of meters). Further, its low power design accompanied by a standardized high-resolution, ranging protocol between peer UWB nodes, makes it a synergistic choice for deployment on the UAV. Thus, our objective is *to localize and track responders (carrying UWB nodes) in real-time even if they are deep indoors, with the help of a UAV (also carrying a UWB node) flying outside*, as shown in Figure 5.1.

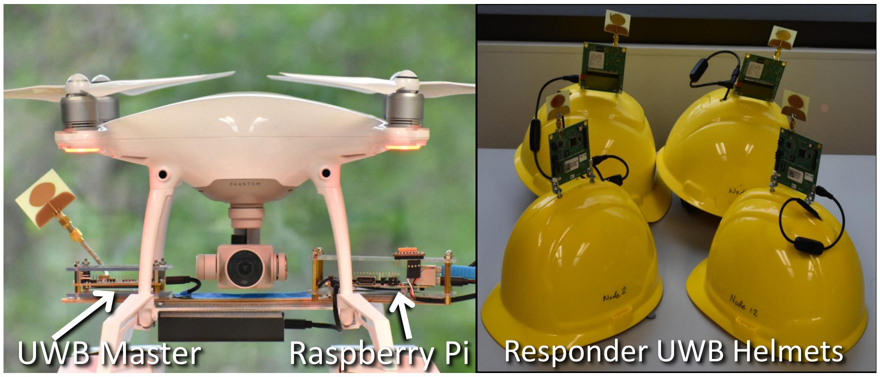


Figure 5.1: *Left*: The TrackIO setup with a DJI Phantom 4 UAV carrying a master UWB node and the Raspberry Pi controller units, *Right*: 4 UWB equipped helmets for first responders.

One might wonder if deploying multiple UAVs to effectively serve as stationary BSs/APs outside can help solve the indoor tracking problem. We argue in Section 5.2.1 that having multiple UAVs outside does not guarantee access to multiple (three or more) of them by a given indoor responder, not to mention the need for their synchronization. More importantly, we show that when available, multiple UAVs need to be efficiently deployed to cover (localize) responders in different sections of the building simultaneously, rather than to serve as stationary APs/BSs. Thus, at the core of our problem, we must *localize and track indoor responders in real-time by a single UAV using its key degree of freedom, namely mobility*. This in turn poses several non-trivial challenges. *(i) Mobility of responders:* The mobility of the UAV is used to create a synthetic aperture *over time*, which serves to provide reference points for localizing an indoor node[1] through multi-lateration. However, such temporal dependency, makes multi-lateration approaches fail significantly (accuracy of about 10m, Section 5.3), when the indoor node is also mobile. *(ii) Indoor coverage:* While UWB's penetration capabilities are better than mmWave, they are still limited to tens of meters and hence cannot guarantee reachability to all nodes. While deploying multiple UAVs, outside different sections of the building can alleviate coverage issues, it still cannot ensure reachability to those that are deep inside the building. *(iii) Real-time tracking:* The UWB protocol provides the basic two-way ranging primitive between a UWB node-pair (UAV and responder in our case). However, employing its TDMA operational structure to collect sufficient ranging measurements to all UWB nodes from the UAV will not be scalable for real-time tracking in a large network. *(iv) Absolute location fix:* Since the UAV localizes the responders with respect to its own position, to get their absolute location fix, we need to accurately estimate the UAV's position as well. Whereas high-end UAVs employ multiple GPS receivers along with inertial sensor fusion to provide position accuracy to under a meter, lower-end UAVs provide accuracies of only around 2-3 m, thereby limiting the accuracy of the overall system.

Towards addressing these challenges, we build TrackIO – a UAV-UWB based system that is capable of localizing and tracking mobile responders to within 1-2 m accuracy from a single UAV outside in real-time, even in deep indoor environments. When multiple UAVs are available, TrackIO deploys them on different sections of the building for wider, simultaneous coverage. In realizing this, TrackIO incorporates four novel elements in its design.
(i) Trajectory Tracking: TrackIO adopts a first-principles approach to directly estimate the trajectory of the mobile node, rather than just its location. TrackIO analytically instruments multi-lateration formulation to not only estimate the location but also the velocity of the responder. It incorporates intelligent mechanisms for adaptively varying the size and choice of the synthetic aperture (anchor points used for localization) to address responders with non-uniform velocity (e.g. those turning corners, etc.).
(ii) Multi-hop Localization Paradigm: TrackIO enables a multi-hop localization paradigm

[1] Responders are synonymously referred to as nodes.

for extended indoor coverage, where, responders directly reachable from the UAV ($\texttt{hop}_1$), are localized first. Then, they serve as anchors for localizing nodes ($\texttt{hop}_2$) that are reachable by them but not by the UAV. Nodes are able to dynamically estimate their own hop status based on their reachability to the UAV and overheard ranging messages from neighboring nodes. TrackIO alleviates the deterioration in accuracy over hops (due to iterative localization), by selecting only upstream nodes with accurate location estimates as anchors for downstream localization.
(iii) Concurrent Ranging Protocol: To enable real-time tracking even for a large, multi-hop network of nodes (e.g. big buildings), TrackIO transforms UWB's sequential ranging protocol into an *efficient, concurrent* one. It leverages the broadcast nature of the wireless medium to (a) parallelize the ranging measurements within each hop, and (b) efficiently multiplexes ranging measurements between hops,while also eliminating redundant message transmissions. TrackIO achieves a $3\times$ speed-up, resulting in a location update frequency of 6 Hz that allows for real-time tracking.
(iv) Reverse Location Look-up: Instead of the UAV serving as the anchor, TrackIO now estimates the location of the UAV itself, by leveraging UWB again. It accomplishes this by using four static UWB beacons, deployed on the roof corners of a responder service vehicle, as anchors. One of these UWB beacons is also fitted with a GPS receiver, whose stationary estimates over time are highly accurate. This coupled with known inter-beacon distances, allows for accurate localization of the UAV to within a meter despite mobility.

We have built a complete version of TrackIO using a DJI Phantom 4 [59] as the UAV, and Decawave DW1000 [48] as the UWB node. The ranging estimates collected at the UAV are transferred to a ground service vehicle, where TrackIO's algorithms estimate the position and trajectory of all the responders in real-time. Our real-world deployment and evaluation across multiple floors of a mid-size office building (2500 sq.m.) reveal that TrackIO is able to track indoor static nodes with a median accuracy of about 1–1.5 m and mobile (even running) nodes with a median accuracy of 2–2.5 m. A demo of TrackIO is available at `http://www.nec-labs.com/trackio`.

While TrackIO leverages UAV and UWB as its modality for enabling real-time tracking of first responders, we would like to note that TrackIO's core mechanisms of trajectory tracking and multi-hop localization can be equally applicable to other localization modalities (e.g. WiFi) as well. Hence, TrackIO's contributions can also benefit other potential indoor localization and tracking applications.

5.2 CHALLENGES IN BUILDING A PRACTICAL SYSTEM

The UAV flies outside to create a synthetic aperture of anchor points, from where it ranges with each of the indoor nodes using UWB, thereby allowing for their subsequent localization through multi-lateration. Albeit straight-forward in principle, realizing this in practice faces several challenges, some fundamental, and others practical that we now outline.

5.2.1 Impact of Responder Mobility

Fig. 5.2(a) shows a typical synthetic aperture where three representative $< location, range >$ tuples are chosen to solve for a stationary node's location using trilateration. The solution is reasonably accurate as all the three $< location, range >$ tuples are consistent with respect to a unique location of the stationary node. Contrast this with figure 5.2(b), where the node moves with an uniform velocity. In this case, the three $< location, range >$ tuples are no longer consistent with respect to any particular node location. Different portions of the synthetic aperture now correspond to different node locations. In other words, the node has changed its position significantly by the time the UAV started and completed building the aperture. This can affect localization accuracy by as much as 10 m, as shown in Fig. 5.8.

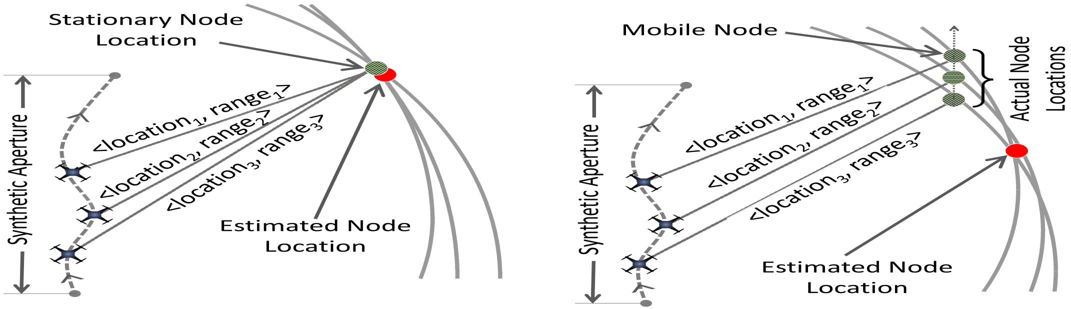


Figure 5.2: (a) Localization of a static node through trilateration (b) Naive trilateration fails for mobile nodes

Can we alleviate the impact of mobility? A natural approach is to figure out if multiple $< location, range >$ tuples can be gathered from distinct UAV locations "simultaneously".

<u>*Using multiple UAVs:*</u> Multiple UAVs form a spatial aperture that can simultaneously collect range estimates in principle. This is however, difficult to realize in practice for the following reasons: (a) It is unlikely that a particular indoor node is simultaneously reachable from multiple UAV locations, (b) Synchronizing the different UAVs as well as their corresponding range estimates in real-time becomes extremely challenging, and (c) Operating multiple UAVs in close vicinity requires sophisticated path planning to be done apriori. Multiple UAVs have a role to play in the broader system (for improving building coverage,

as we discuss later). However, they are less useful to solve the problem of node mobility, which motivates us to address the problem with a single UAV.

Increasing the UAV's speed relative to responders: Another approach to counter node mobility can be to increase the UAV's velocity. Figure 5.3 shows the limited benefit of moving the UAV faster. Even when the UAV is traveling at 10m/s it cannot completely compensate for the node's mobility. Moreover, moving the UAV too fast causes the channel to change very rapidly, resulting in ranging errors.

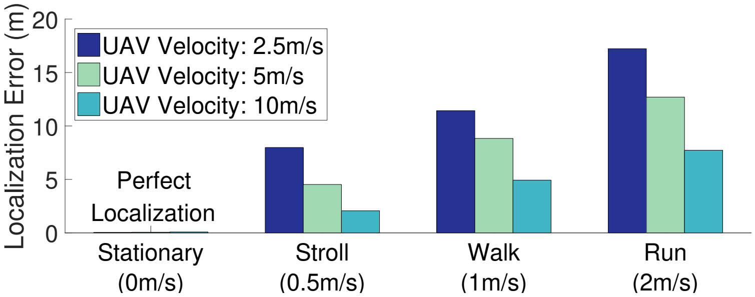


Figure 5.3: Localization accuracy improves with UAV speed, yet falls short of the target accuracy.

5.2.2 Insufficient Building Coverage

The FCC power emission limit for UWB transmitters is – 41.3 dBm/MHz [42] that severely restricts the communication range between two UWB nodes. With the UAV located outside the building and limited indoor penetrability, some nodes that are relatively deep indoors are not directly reachable. We perform elaborate measurement studies to characterize the communication range in such indoor environments. Figure 5.4 shows the packet-loss percentage as the distance between the nodes increases in a cluttered indoor space. Such ranges could vary from about 30 m (50% loss) in very dense/cluttered indoor environments (e.g., rooms with concrete walls) to about 60 m in relatively open indoor spaces (e.g., office, library, shopping malls etc.).

Multiple UAVs can improve but not solve the coverage problem. With the UAV flying approximately 10 m away from the face of the building, nodes that are about 30 m inside are directly reachable. This limits the indoor coverage area to a great extent. Note that even flying multiple UAVs along the four faces of a medium-sized building (floor area $\approx$10000 sq.m) only improves coverage in the building's periphery but not in the deep interiors that account for about 20% of the indoor space (see figure 5.5). Given the criticality of the application, complete indoor coverage is of paramount importance. This necessitates the design of a multi-hop localization network, where range estimates and hence localization can be achieved from nodes of a given hop to nodes belonging to the next hop

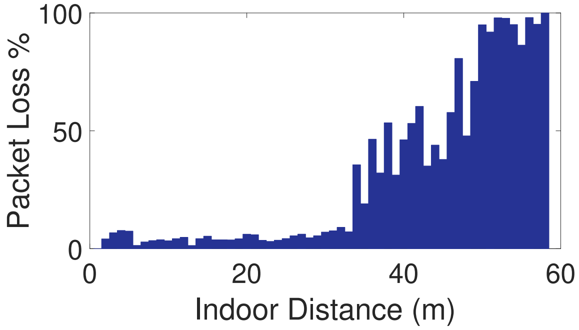


Figure 5.4: UWB packet-loss over various distances in a cluttered office environment.

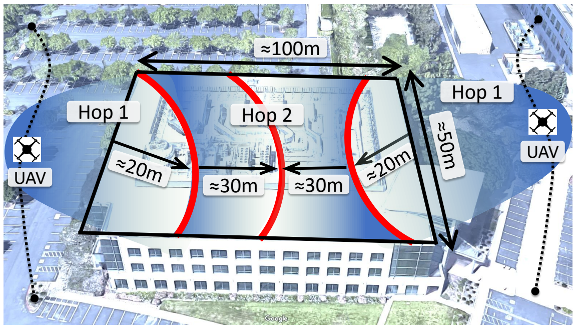


Figure 5.5: UAVs on the outside may not be able to cover nodes deep indoors. A multi-hop solution is necessary.

and so on.

Challenges with multi-hop localization. Realizing multi-hop localization is challenging for several reasons: (i) iterative localization leads to cascading errors and hence poor accuracy across hops; (ii) nodes need to identify their reachability status (e.g $\texttt{hop}_1$, $\texttt{hop}_2$, etc.) to other nodes to help track a dynamic, multi-hop topology; (iii) orchestration of ranging measurements across hops becomes critical for ensuring real-time tracking of the multi-hop network.

5.2.3 Inability to Track Real-time

In a large network of nodes spanning multiple hops, a *time division* (TDMA) scheme needs to be designed that runs TWR across relevant pairs of nodes to estimate their range fast enough to relatively localize all nodes in the network. Clearly, executing a TWR across all pairs of nodes is not suitable: unreachable links will waste time, and in a size N network, one round of (range) data collection will require $O(N^2)$ time slots. Hence, for a network consisting of several tens of nodes, collecting a single set of range information might take several hundreds of milliseconds. With several such sets needed to position the indoor nodes, the total delay can be several seconds. Further, with the mobility of nodes resulting in highly dynamic topologies, it becomes very challenging to track nodes in real-time with such an update rate of measurements.

5.2.4 Imprecise UAV Localization

Note that the UAV's location measurements need to be as precise as possible in order to leverage the highly precise range estimations offered by the UWB technology. Unfortunately, UAV location estimates obtained out-of-the-box is at least an order of magnitude less precise compared to UWB ranges. For instance, consider the location estimates obtained from a GPS device. In an open field, such locations have minimal errors (≈ 2–3m). However, in scenarios, where the UAV moves along the periphery of a building, the GPS signal reception can be significantly hampered resulting in the error to escalate to as high as 15–20m [60].

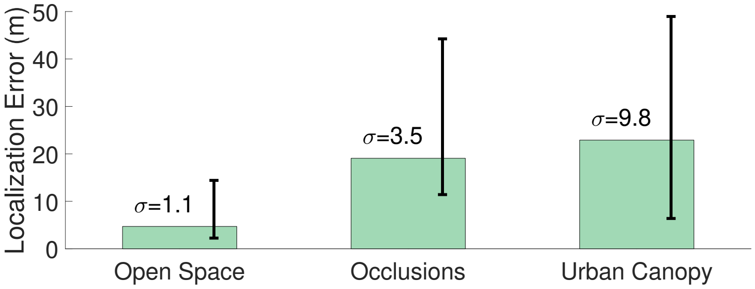


Figure 5.6: Effect of UAV's GPS errors on localization accuracy

In figure 5.6 we show the impact on localization accuracy of a static node using simple trilateration in three different deployment settings; from a relatively open space to locations having partial occlusions and urban canopies. Note that even for a static outdoor node, slightly erroneous GPS locations of the UAV can be detrimental for its eventual localization. Assuming GPS corrections and inertial sensor fusion applied by the UAV, the errors could be at best, $1 - 2\,\text{m}$ even when the node is outside and static; localizing a mobile node indoors would only lead to significantly degraded accuracies.

5.3 SYSTEM DESIGN

We now present TrackIO—a UAV-UWB based system that is capable of localizing and *tracking mobile* responders to within 1-2 m accuracy from a single UAV outside in *real-time*, even in *deep indoor* environments. TrackIO accomplishes this without the necessity or dependence of any infrastructure deployed indoors. TrackIO can almost instantly be functional from the time of launch (under a minute). This is achieved by employing a host of algorithmic and architectural changes to the underlying multilateration and ranging protocols.

Overview

Fig. 5.7 shows a snapshot of TrackIO in action along with its various architectural components. The UAV flying outside the building's periphery is equipped with a UWB *master node* that collects range information from the *client nodes* inside the building. The nodes are possibly worn by personnels (e.g., firefighters, military troops, emergency responders etc.) who are tracked through our system. Client nodes that are directly reachable from

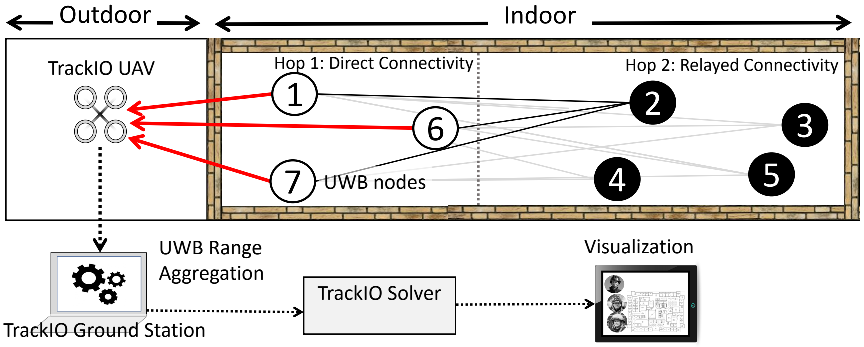


Figure 5.7: High level system design

the UAV's master node are designated as $\texttt{hop}_1$ nodes, additional nodes are referred to as $\texttt{hop}_2$ nodes, $\texttt{hop}_3$ nodes and so on based on subsequent reachability. In Fig. 5.7, the UAV directly ranges the nodes in $\texttt{hop}_1$, who in turn range the nodes in $\texttt{hop}_2$ and relay information back to the UAV. The UAV offloads range information to a ground control station that solves for the locations of all client nodes. Additionally, we develop a mobile application that can be used to visualize the tracking information on a map with sub-second latency.

When multiple UAVs are available, they are deployed on different sections of the building and/or at different altitudes of the same section (for tall buildings) for wider, simultaneous coverage. Since each UAV would execute TrackIO in parallel, we focus on a single UAV's operation in the rest of this section. Also, for easier exposition, we focus on the UAV localizing responders in a single floor (horizontal plane) by fixing its altitude appropriately. How the UAV scans floors and identifies the appropriate altitude (z^*) is covered in Section 5.3.5. Some results presented in this section are obtained from simulation studies, which are intended for highlighting the intricate aspects of our system design. Nonetheless, sections 5.4 and 5.5 present extensive evaluation results from experiments carried out in real testbeds.

5.3.1 Tracking Trajectory of Mobile Nodes

Estimating Velocity through Synthetic Aperture

Recall that when a node is mobile, the $< location, \ range >$ tuples measured by the UAV do not uniquely map to a single location, resulting in poor localization accuracy of multilateration solvers. Instead of alleviating the impact of mobility, TrackIO adopts a first-principles approach to directly estimate the trajectory (speed and heading) of the mobile node, rather than just its position. To accomplish this, TrackIO analytically instruments the multilateration formulation to estimate both the initial location (x, y) as well as the velocity vector $(V_x\hat{i} + V_y\hat{j})$, where $\hat{i}$ and $\hat{j}$ are unit vectors along positive X and Y axes respectively of the node. Using these, the node's traversed path can be traced. This assumes that human mobility can be approximated with uniform velocity, which is reasonable within the short time-scales (few seconds) of the UAV's synthetic aperture. This assumption is relaxed in Section 5.3.1, where we show how non-uniform mobility (e.g. turning corners, etc.) can also be addressed in this framework.

Suppose we have ranging measurements from n consecutive UAV locations – n is called the aperture size and is essentially a moving window of n historical measurements. For any time instant T_i ($i = [1..n]$), the UAV records the mapping $< location_i,\ range_i >$, where $location_i$ is the UAV's 3D location and $range_i$ is the distance estimate of the mobile node from the UAV. The mobile node is located at an unknown location $(x_i,\ y_i,\ z_i)$. We denote the UAV's 3D-location as $(Cx_i,\ Cy_i,\ Cz_i)$. The measured range is given by:

$$range_i = \sqrt{(Cx_i - x_i)^2 + (Cy_i - y_i)^2 + (Cz_i - z_i)^2} \tag{5.1}$$

Assuming we know which building floor the responder is currently occupying, we do not need to solve for z_i $(= z^*)$. Yet, this is a single equation with two unknowns—we cannot directly solve for $(x_i,\ y_i)$. Even if we obtain multiple such ranges, each equation will add a new set of unknowns. However, the new unknowns are not independent, but related through the node's velocity. Hence, assuming the node is moving at a constant velocity, there are inherently only four unknowns $(x_1^*,\ y_1^*,\ V_x^*,\ V_y^*)$ that do not increase with additional ranges, thereby allowing us to solve for them.

We can reformulate this as an unconstrained minimization problem that attempts to find the best fit, i.e. location and velocity that minimize the following error function:

$$(x_1^*,\ y_1^*,\ V_x^*,\ V_y^*) = \arg \min_{(x_1,\ y_1,\ V_x,\ V_y)} f$$

$$f = \sum_{i=1}^{n} ((Cx_i - x_i)^2 + (Cy_i - y_i)^2 + (Cz_i - z_i)^2 - range_i^2)^2 \tag{5.2}$$

The various $(x_i,\ y_i)$ are obtained from the intial $(x_1,\ y_1)$ and velocity (V_x, V_y) based on kinematic equations:

$$
\begin{aligned}
x_i &= x_{i-1} + V_x \cdot \Delta T_i = x_1 + V_x \sum_{j=1}^{i} \Delta T_j \\
y_i &= y_{i-1} + V_y \cdot \Delta T_i = y_1 + V_y \sum_{j=1}^{i} \Delta T_j
\end{aligned}
\tag{5.3}
$$

where ΔT_i denotes the time between measurements. Since $(x_i,\ y_i)$ are generated based on the initial location $(x_1,\ y_1)$ *and* the velocity vectors $(V_X,\ V_y)$, by minimizing equation 5.2, we obtain the closest approximation of both, location and velocity vectors for the node. The first output from this solver is obtained only after n measurements (typically a few seconds worth of data) have been recorded. Thereafter, a location update is obtained for every round of range measurements. Thus, the system's steady state update rate depends only on the duration of one range measurement round, and does not depend on the aperture size.

We now analyze the improvement in localization achieved by incorporating velocity vectors over simple multilateration. Our simulation framework mimics a UAV and a set of indoor UWB nodes that follow predetermined trajectories at any desired speed. We introduce an empirically derived range estimation error to the ranges. Fig. 5.8 shows how simple multilateration results in higher localization errors with increasing node velocities. The UAV is assumed to move at a fixed 5 m/sec velocity. Note that even for human walking speeds the error could be as high as 10 m. On the contrary, the velocity-based solver is least impacted by increasing velocity of the mobile node.

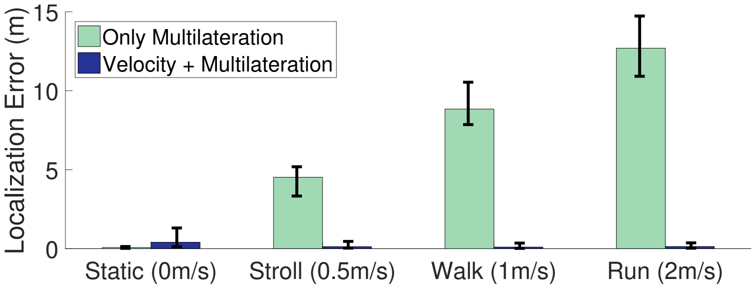


Figure 5.8: Localization error remains minimal when using velocity vectors even under fast human mobility

Adaptive Apertures for Non-uniform Velocity

The above approach assumes that the node does not change its velocity (speed or direction) during the course of one aperture window (say, 4 secs). However, this assumption is broken if the node turns, accelerates or halts. In principle, this could be solved by adding higher order derivatives of the node's location (e.g., acceleration, jerk) into the kinematics equations employed by our location solver. However, our analysis indicates that such an approach is rather contrived. It makes the solver prone to overfitting and extremely sensitive to range errors. Further, given the short time scale of the aperture window, we find that the approximation of uniform velocity does not hurt the performance much during acceleration and halting but does induce significant errors during *turns*, which we now address. We propose to utilize the solver's confidence in the estimated location to infer non-uniform velocity and when detected, trigger an aperture reset that eliminates measurements prior to the turn.

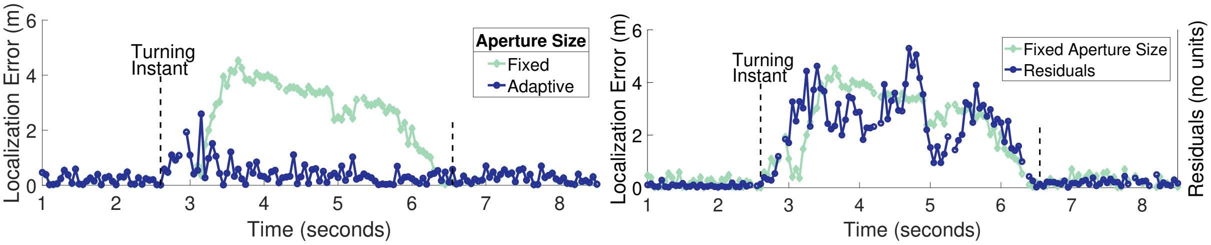


Figure 5.9: (a) Localization error is high during turns. Resetting the aperture helps curtail the loss in localization accuracy. (b) The high error-residual also indicates low solver confidence in the location estimate providing a hint for turn detection.

Impact of turns. Fig. 5.9(a) shows the impact of sudden turns on the localization error (green line). We simulated a node moving in a straight line, then taking a 90° turn, and continuing again in a straight line. An aperture of 4 seconds—UAV locations and the corresponding ranges of the past 4 seconds—are used to estimate the current node location. Observe how the localization error (grey line) starts to increase from the point where the aperture's *head* crosses the turning position (first dashed vertical line) and falls back to its pre-turn values after the aperture's *tail* has crossed the turning position (second dashed vertical line).

Adaptive aperture to address turns. If we have a mechanism to detect turns, we could potentially eliminate historic measurements till the turn and restart constructing the aperture. To understand the benefit, we introduce the notion of an adaptive aperture in the above simulation. At the time of the turn, we remove all history and restart estimating location after a short history has built up[2]. Just after resetting history, the localization error is indeed high (dark blue line just after the "turning instant" in Fig. 5.9(a)) but quickly

[2] During the short period that new history is being built, the system continues to output results from the previous aperture.

recovers and becomes acceptable once the aperture fills up with relevant measurements after the turn. In comparison, if a fixed aperture size is used, the effects of a turn last for the entire duration of the aperture (green line).

Triggering an adaptive aperture. During turns, the solver is unable to provide a reasonable answer since no single velocity estimate can represent all the measurements. This results in larger residual errors after solving Equation 5.2. Observe in Fig. 5.9(b) that the solver's residuals (in arbitrary units) are highly correlated with localization error. Thus, a sudden increase in the residuals helps identify non-uniform velocity events such as turns. We use Gaussian Mixture Models on the residuals to identify a changing trend in them and captures such events.

In summary, localization of mobile nodes, even those with non-uniform velocity is possible through a combination of joint location-velocity solving and by adaptively resetting the aperture size. At any given instant, our solver uses different aperture sizes that are appropriate for each node.

5.3.2 Multi-hop Tracking for Coverage

TrackIO is designed to function even if some nodes are beyond the UAV's direct range. TrackIO allows such unreachable nodes to range with other nodes in the vicinity which can in-turn reach the UAV and/or have already been localized. Thus, a multi-hop topology is dynamically created with nodes belonging to different hops based on their reachability characteristics. The UAV's synthetic aperture localizes first hop (directly reachable) nodes. These $\mathtt{hop}_1$ nodes then act as anchors for localizing $\mathtt{hop}_2$ nodes. This process iterates across hops. TrackIO employs several mechanisms to ensure that mobile nodes can be accurately localized even across multiple hops.

Dynamic estimation of hop membership. Nodes that are within the UAV's UWB communication range, directly receive ranging messages initiated by the UAV, and classify themselves as $\mathtt{hop}_1$ nodes. Those that do not receive messages from the UAV but receive some of the response messages sent by $\mathtt{hop}_1$ nodes, classify themselves as $\mathtt{hop}_2$ nodes and so on. Thus, nodes can determine their own hop membership in a decentralized manner.

Anchor selection for iterative localization. Two components contribute to the final localization error of $\mathtt{hop}_\mathtt{m}$ nodes: 1. relative localization error of $\mathtt{hop}_\mathtt{m}$ nodes with respect to

$\texttt{hop}_{\texttt{m-1}}$ nodes, and 2. localization error of $\texttt{hop}_{\texttt{m-1}}$ nodes chosen as anchors. Without loss of generality, $\texttt{hop}_{\texttt{m-1}}$ nodes can be assumed to be spaced far apart compared to the synthetic aperture formed by the UAV. This increased spacing between anchors, *improves* $\texttt{hop}_{\texttt{m}}$ localization, compared to that of $\texttt{hop}_{\texttt{1}}$ nodes (w.r.t. the UAV). However, the localization error of the $\texttt{hop}_{\texttt{m-1}}$ nodes and recursively that of upstream hop nodes, will cumulatively contribute to the error of $\texttt{hop}_{\texttt{m}}$ nodes. Thus, the choice of anchors in $\texttt{hop}_{\texttt{m-1}}$, has a cascading impact on the localization accuracy of downstream nodes (i.e. hops $\geq m$).

Nodes that are static or moving with a uniform velocity in $\texttt{hop}_{\texttt{1}}$ inherently have better localization accuracy than those with non-uniform velocity. Hence, by leveraging the solver's ability to identify such nodes (those with high residuals), TrackIO avoids selecting them as anchors for localizing $\texttt{hop}_{\texttt{m}}$ nodes, curtailing the cascading effect across hops.

Instantaneous mobility tracking beyond $\texttt{hop}_{\texttt{1}}$. In contrast to the first hop nodes which are localized through a *temporal* aperture created by the UAV's motion, $\texttt{hop}_{\texttt{m}}$ ($m > 1$) nodes are localized through a *spatial* aperture formed from a diverse placement of $\texttt{hop}_{\texttt{m-1}}$ nodes. This decoupling (from UAV's mobility), allows for *instantaneous* localization of $\texttt{hop}_{\texttt{m}}$ nodes from previously obtained $\texttt{hop}_{\texttt{m-1}}$ locations. The time scale of such localization is in milliseconds within which the nodes move a negligible distance. As a result of the spatial aperture employed, $\texttt{hop}_{\texttt{m}}$ ($m > 1$) nodes can use conventional multilateration approaches (without need for velocity vectors) even when they are mobile.

Localizing $\texttt{hop}_{\texttt{1}}$ nodes with non-uniform velocity using downstream spatial apertures. Unlike $\texttt{hop}_{\texttt{1}}$ nodes, mobility is not a concern for $\texttt{hop}_{\texttt{m}}$ ($m > 1$) nodes as they are instantaneously localized using a *spatial* aperture formed from high-confidence (low residual) $\texttt{hop}_{\texttt{m-1}}$ nodes. Hence, $\texttt{hop}_{\texttt{2}}$ nodes can in-turn, form a spatial aperture (serve as anchors) and correct the location of $\texttt{hop}_{\texttt{1}}$ nodes, which are currently experiencing non-uniform velocity (low confidence, high residual). Fig. 5.10 shows the localization accuracy of a turning $\texttt{hop}_{\texttt{1}}$ node using ranges from $\texttt{hop}_{\texttt{2}}$ nodes. Observe how the turn gets localized precisely using the spatial aperture from $\texttt{hop}_{\texttt{2}}$. Thus, TrackIO is able to eliminate most of the impact of non-uniform velocity of $\texttt{hop}_{\texttt{1}}$ nodes.

Note that this downstream spatial aperture technique is opportunistic – it can be used when enough $\texttt{hop}_{\texttt{2}}$ nodes exist. In contrast, the mechanism of adaptive (temporal) aperture of the UAV described in Section 5.3.1, provides benefits even when no other nodes

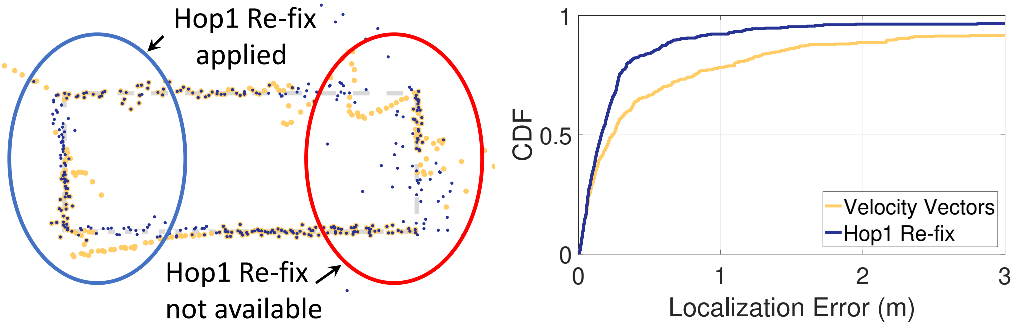


Figure 5.10: Re-fixing $\texttt{hop}_1$ nodes using $\texttt{hop}_2$ nodes improves localization even during turns. But, it may not be always available.

exist in the topology. Therefore, TrackIO incorporates both these techniques to address non-uniform velocity of $\texttt{hop}_1$ nodes.

Leveraging multiple UAVs vs. multiple hops. While multi-hop localization allows for coverage of even large buildings using a single UAV, the localization error of its downstream hop nodes (i.e. $m > 2$) will increase and might not satisfy our desired target of 1-2m. Hence, TrackIO leverages the multi-hop paradigm to primarily reach deep interiors of buildings (where even multiple UAVs cannot help), while employing multiple UAVs to provide non-overlapping, peripheral coverage for large buildings. Thus, using a combination of multiple UAVs and hops, TrackIO is able to cover large buildings with just two hops from a single UAV.

Handling $\texttt{hop}_2$ disconnections. In rare circumstances, if a node goes out of range from all hop-1 nodes, its localization must rely on an IMU-based dead reckoning system. This approximate location estimate will then be communicated with the UAV using alternative communication modes (such as WiFi or cellular data). Adding support for such eventualities is left to future work.

5.3.3 Concurrent Ranging

A fast and reliable ranging protocol is essential to create a real-time localization system. Since the UWB ranging protocol is designed for ranging between a pair of nodes, it does not broadcast messages. This leads to a sequential ranging of every node in a hop, which is not scalable for real-time operation, especially in a multi-hop network. The key idea in TrackIO is to leverage the broadcast nature of wireless signals to communicate and hence *concurrently* range with multiple nodes using a single transmission. To this end, TrackIO makes appropriate modifications to the underlying protocol (Fig. 3.2) to create a concurrent ranging scheme (Fig. 5.11). We describe the scheme for the first two hops; subsequent hops are similar.

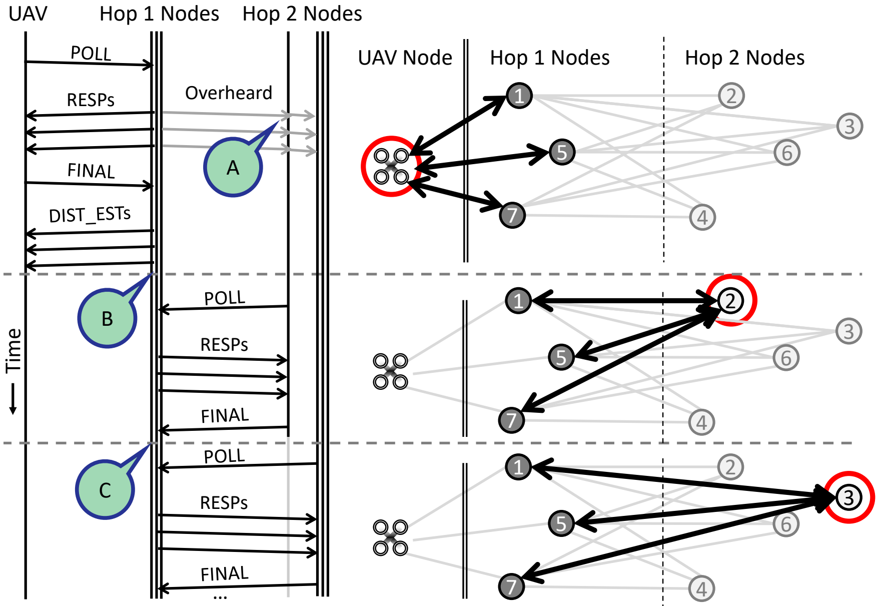
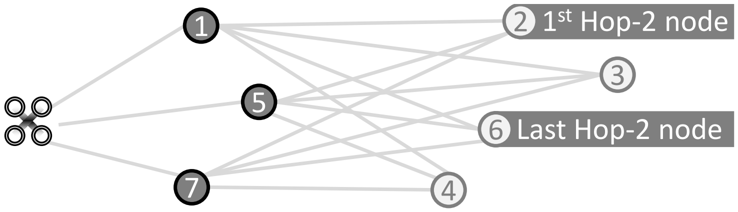


Figure 5.11: Progress of the protocol in a 2-hop example topology.

Concurrent ranging at $\texttt{hop}_1$**.** The UAV *simultaneously* initiates ranging with all reachable $\texttt{hop}_1$ nodes by broadcasting a single POLL. Each node that receives this message, takes turns (based on its hard-coded NodeID) to send a RESP message. After collecting the timings from all the RESPs, the UAV broadcasts a single FINAL message containing information for all $\texttt{hop}_1$ nodes. On receiving the FINAL, all $\texttt{hop}_1$ nodes calculate their distance from the UAV and send it back to the UAV (DIST_EST messages).

Concurrent ranging at $\texttt{hop}_2$**.** Identical to $\texttt{hop}_1$ nodes, $\texttt{hop}_2$ nodes listen to the channel for messages. However, being outside the direct communication range of the UAV, they cannot receive the POLL message. Instead, they only overhear the messages sent by nearby $\texttt{hop}_1$ nodes in response to the UAV's POLL (point A in Fig. 5.11). After all $\texttt{hop}_1$ nodes have completed sending their DIST_EST messages (point B in Fig. 5.11), the first $\texttt{hop}_2$ node initiates a full sequence of POLL-RESPs-FINAL *simultaneously* with all $\texttt{hop}_1$ nodes in the vicinity. $\texttt{hop}_2$ nodes follow the same protocol as the UAV with one subtle difference. $\texttt{hop}_1$ nodes do not send DIST_ESTs back to $\texttt{hop}_2$ nodes. Instead, $\texttt{hop}_1$ nodes calculate and locally store all the $\texttt{hop}_1$-$\texttt{hop}_2$ ranges, which are piggybacked on the subsequent DIST_EST message. This saves unnecessary network overhead, speeding up the collection of range estimates. All $\texttt{hop}_2$ nodes take turns (point C in Fig. 5.11), followed by the UAV starting the next round.

Efficient multiplexing of ranging between hops. Initially the UAV is not aware of the topology. Hence, it waits for all the nodes in the network to send a RESP. Once it has received the last RESP (or, after a timeout), the UAV creates a bitmap (Fig. 5.12) indicating which nodes are deemed to be in $\texttt{hop}_1$ (setting the corresponding bit to one) based on the responses. The UAV sends this bitmap in its FINAL message. When $\texttt{hop}_1$ nodes send a DIST_EST message it also contains a copy of this bitmap. A node that receives such a DIST_EST, but not the POLL from the UAV, would see its bit cleared and know that it belongs to hop2. Also, it would know how many other $\texttt{hop}_1$ nodes are expected to send their DIST_ESTs, and the order of all other $\texttt{hop}_2$ nodes. This allows $\texttt{hop}_2$ nodes to efficiently take transmission turns without collision, even when they are not in communication range of each other and the UAV. The UAV generates the bitmap dynamically in every round to track topology dynamics due to node mobility.



Node ID	7	6	5	4	3	2	1
Bitmap	1	0	1	0	0	0	1

= Three $\texttt{hop}_1$ nodes

Figure 5.12: The bitmap constructed by the UAV and sent in the FINAL message enables a collision free $\texttt{hop}_2$ communication.

Finally, the variable length DIST_EST sent by $\texttt{hop}_1$ nodes piggybacks their distance from all $\texttt{hop}_2$ nodes obtained in the *previous* round, along with their own *current* range estimates to the UAV. The UAV aggregates all the information received in the DIST_ESTs and forwards to a ground control center for further processing.

5.3.4 Reverse Lookup for UAV Location Fix

Obtaining the UAV's precise GPS location is critical to TrackIO's end-end accuracy. This can be challenging since off-the-shelf GPS receivers have multi-meter location errors [60, 61]. High-End UAVs already employ GPS chips with better precision and higher update rate [62], and improve the precision further by incorporating IMU data as well. Some UAVs [62] also support custom, albeit expensive RTK solutions [63] that promise location accuracy within a few cm, but require precise GPS transmitters in the vicinity.

In cases, where such precise UAV location estimates are not possible, TrackIO leverages UWB to also localize the UAV. It places four *static* UWB nodes as anchors at known locations on the ground. One of these anchors is also fitted with a GPS receiver. The known, exact, pairwise distances between the anchors enables TrackIO to accurately determine the GPS coordinates of all the static anchors. These static anchors in turn allow for accurate localization of the UAV itself. We envision that these anchors can be permanently mounted at the four corners of a service vehicle (at different heights to provide vertical diversity). The service vehicle can use sophisticated GPS techniques [64, 65, 66, 67, 39, 63] to achieve better accuracy for the ground anchors.

5.3.5 TrackIO's Operations in a Nutshell

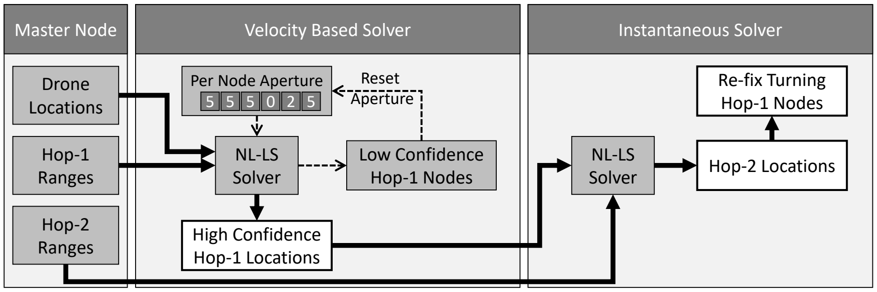


Figure 5.13: TrackIO System Design

When a UAV is launched to cover a section of the building, the static UWB anchors on the ground start localizing the UAV to get its precise location estimate. The UAV performs its flight trajectory to start creating a continuous, moving window of synthetic apertures (of 4 secs each). Within each of its aperture window, it performs the following. It executes its concurrent ranging protocol to help classify nodes into various hops based on their reachability. It then localizes the $\texttt{hop}_1$ nodes first using its location-velocity solver that estimates both the location and velocity. Using the error residuals of the solver, it employs only high-confidence $\texttt{hop}_1$ nodes (static or uniform velocity) as anchors for localizing the $\texttt{hop}_2$ nodes. For the latter, it employs a conventional multi-lateration solver to obtain only the location of $\texttt{hop}_2$ nodes, which being instantaneous, is sufficient. Finally, it uses a spatial aperture of $\texttt{hop}_2$ nodes (anchors), along with an adaptive (temporal) aperture from the UAV, to refine the location estimate of the $\texttt{hop}_1$ nodes that have non-uniform velocity.

Altitude Considerations. So far, we have only focused on the horizontal plane and assumed that TrackIO is aware of the nodes' altitude. However, in cases, when TrackIO is not aware of the floor where the service personnel currently are, the horizontal localization error can be significant (since the algorithm will not take into account the additional vertical offset the signals have to travel). We address such situations by detecting the appropriate altitude (and hence floor) through a special one-time maneuver of the UAV. We move the UAV up and down through a short vertical distance that spans the target floors. During this movement, as the UAV approaches the horizontal plane of the nodes, its range estimates to the nodes should start to decrease, reach a minimum when it is on the plane, and increases when it moves away from the plane. TrackIO records the altitude (z^*) as that corresponding to the minimum range estimates and hence determines the floor of interest. TrackIO then uses this altitude to execute its localization process for the target floor.

5.4 IMPLEMENTATION AND TESTBED SETUP

We build a custom payload consisting of a Decawave DW1000 UWB module and a Raspberry Pi 3 used as the TrackIO controller. The payload, weighting about 200 grams is mounted onboard a DJI Phantom 4 UAV platform. A fully charged UAV flight with our current prototype lasts for about 20 mins (≈25 mins without payload). In the following we describe the key hardware/software components that form TrackIO.

5.4.1 TrackIO Components

UWB Modules: The UWB module mounted on the UAV acts as the master node and is responsible for collecting ranging information from the client nodes. Alongside the DW1000 RF chip, the UWB module houses an ARM based microcontroller that runs our multi-hop ranging protocol (implemented in about 3000 lines of C code). The latter collects inter-node ranging information (at about 6 Hz) which is read by the controller Raspberry Pi and forwarded to the ground station through a WiFi interface.

Ground Station: Ground station refers to the compute node responsible for collecting ranging information obtained from the UAV and running TrackIO's localization algorithms. First, it localizes the UAV using the four fixed client nodes on the ground. These nodes are placed at different heights (vertical diversity) on four vertices of a $5\,\mathrm{m} \times 5\,\mathrm{m}$ square to emulate a service vehicle housing the ground station. One node is equipped with a GPS receiver for an absolute location fix. Second, the UAV's location is fed along with the rest of the range information that simultaneously solves all client node locations. We implement the solver algorithms in *Python* that run in real time on the ground station compute node (a Core i7 Lenovo laptop). We also implement an Android application that shows client node locations on a map.

Flight Automation: Automating the flight offers flexibility to programmatically control the flight's trajectory as well as its speed. We use the `Android Mobile SDK` [68] provided by DJI to program two candidate trajectories for our UAV to follow: (a) `STRAIGHT`, a straight line path of length 30 m, and (b) `WAVY`, a sinusoidal path of length 30 m with an amplitude of 5 m (see figure 5.16). Note that such automation also helps us to re-run/repeat flights for controlled experiments, which would have been otherwise impossible in case of manually controlled flights.

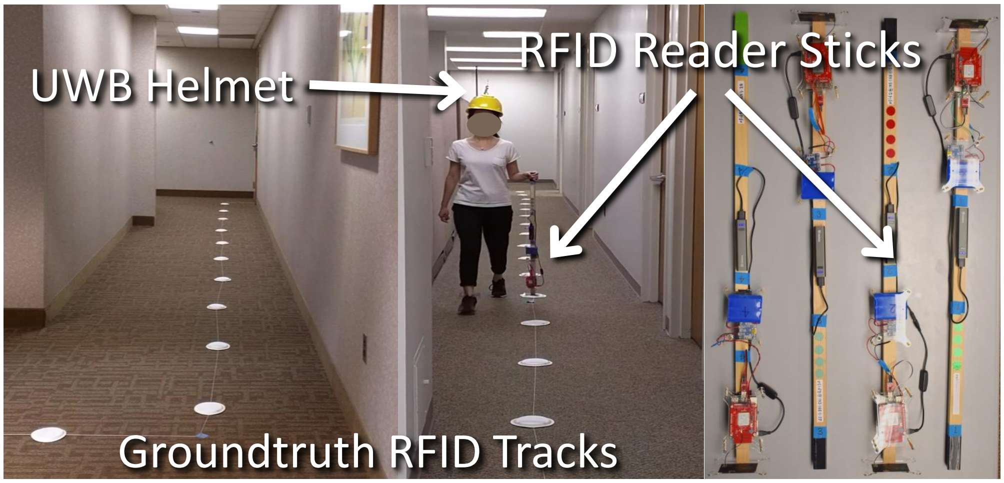


Figure 5.14: Snapshots of trajectories marked with RFID tags. A volunteer is shown walking along the track with the RFID reader stick in her hand.

5.4.2 Testbed Setup

We deploy TrackIO with a single UAV in the third and fourth floor of our department building spanning approximately 1250 sq. meters (half the building's floor area). Nine client nodes are placed indoors that mimic static or mobile first responders. Out of these nine nodes, six are in `hop-1` (3 static, 3 mobile) and remaining three in the `hop-2` (2 static, 1 mobile).

Obtaining location groundtruth: For static indoor nodes, the node location is accurately estimated with the help of a laser ranger. For tracking the mobile nodes' groundtruth positions, we deploy fixed RFID tags on the ground, one every meter, along predefined trajectories. We create portable *RFID reader sticks* equipped with a ThingMagic M6E-Nano readers (Fig. 5.14). We adjust the reader's transmit power as well as the antenna orientation to limit the reading range to about 50 cm. The volunteers are instructed to move along the trajectories while holding the stick vertically. The stick also hosts a Raspberry Pi that controls the reader and logs timestamped entries of RFID tags along with RSS/phase information ($\approx$10 Hz) it reads along the trajectory. We post-process such logs to obtain accurate estimations of position (within 20 cm) and velocity of the mobile node at granular timescales.

Trajectories: We lay out the trajectories within our office area spanning multiple rooms, cubicles, hallways and open spaces. Specifically we construct four different trajectories, three in the first hop and one in the second hop. We create the trajectories with increasing number of turns in them. The first three trajectories are (a)LINE, a linear trajectory of length 20 m, (b)TRIANGLE, a triangular trajectory with a perimeter of 30 m and (c)RECT, a rectangular trajectory with a perimeter of 40 m. The trajectory in the second hop is roughly a 30 m long sinusoidal path (SINU).

5.5 EVALUATION

We present evaluation results from experiments conducted in our real testbed discussed in §5.4.2. Recall that we use 6 nodes (3 static, 3 mobile) in $\texttt{hop}_1$ and 3 nodes (2 static, 1 mobile) in $\texttt{hop}_2$. Four volunteers (mobile nodes) are simultaneously instructed to move along their designated trajectories at different speeds. Combined, we accumulate over 2 hours worth of traces accounting for over $10+$ Kms of total trajectory length. Evaluating TrackIO's performance (w.r.t. groundtruth) through controlled experiments requires us to do trace-driven analysis of the ranging information logged by the ground station compute node. However, our system receiving range information at 6 Hz is capable of *real time* operations. In §5.5.4, we highlight end-to-end latency of our system for various node distributions. Fig. 5.15 shows the median localization error for both $\texttt{hop}_1$ and $\texttt{hop}_2$ nodes, the latter localized using static or mobile $\texttt{hop}_1$ nodes. While static nodes are localized with an accuracy of $1-1.5$ m, note that even for mobile nodes, the median localization error is a little less than 2 m ($\texttt{hop}_1$) to around 2.5 m ($\texttt{hop}_2$). In extreme cases, where the $\texttt{hop}_2$ nodes are localized using all mobile $\texttt{hop}_1$ nodes and the latter do not offer a good spatial diversity (e.g., all $\texttt{hop}_1$ nodes are in close vicinity), TrackIO still offers a localization accuracy of about 4 m (top 10 percentile). However such situations can be avoided by judiciously selecting nodes in hop$_1$ that offer spatial diversity.

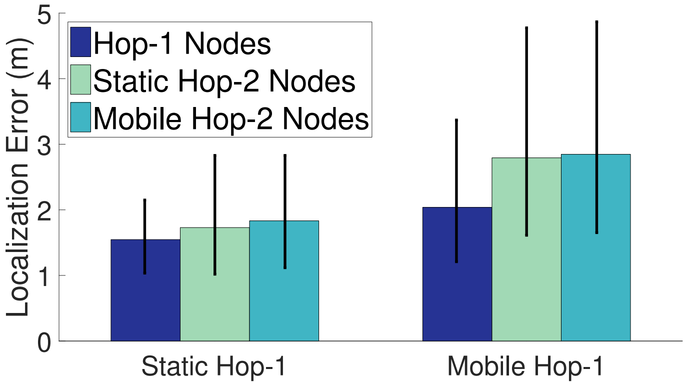


Figure 5.15: Overall localization accuracy for static and mobile hop$_1$ and hop$_1$ nodes

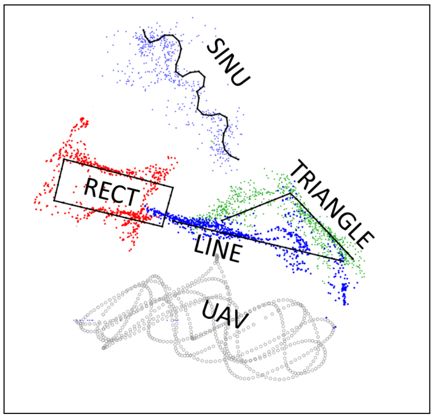


Figure 5.16: Scatter plot of estimated trajectories for all mobile nodes.

5.5.1 Hop$_1$ Localization Performance

We now evaluate the hop$_1$ localization error over the dimensions of node speed, trajectory, and turns.

Effect of node speed. Since TrackIO jointly solves for both location and velocity of each node, ideally, the localization error should be independent of the velocity for nodes moving at a constant velocity. However, when moving at a brisk pace, the human body performs a complex set of movements, including bobbing of the head, which strains the constant velocity assumption. To evaluate these practical limitations, a volunteer moved along LINE at different speeds—a stroll, walk, and a run. Fig. 5.17 shows the resulting localization accuracy. The reported speeds are the average speed obtained from ground-truth. The median localization accuracy is around 1.5 m during the stroll, whereas it is around 2.8 m during the run. The gains over simple multilateration are above $3\times$ for all the velocities.

Effect of trajectory shapes. Fig. 5.18 shows the impact of different trajectory shapes (LINE, TRIANGLE, RECT) on the localization accuracy. While using naive multilateration, the localization errors can spike upto 10 m, TrackIO makes the system resilient to turns (median ≈2 m for all trajectories). Minor differences do exist which can be explained by the increasing number of turns present in the respective trajectories.

Effect of Turns. Human motion mostly comprises of straight lines interspersed with turns of various degrees. Fig. 5.19 shows the localization errors during turning events versus that during traversal in straight line segments. Note that the errors can significantly spike

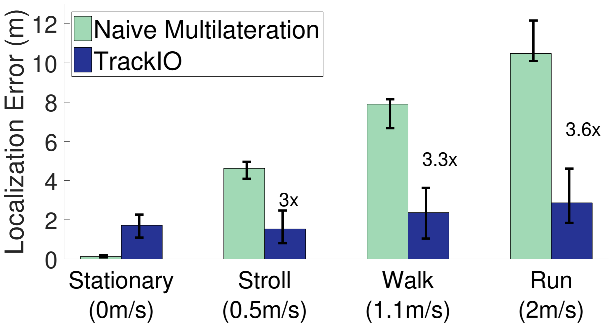


Figure 5.17: TrackIO gains significantly over simple multilateration by solving for velocity vectors as well.

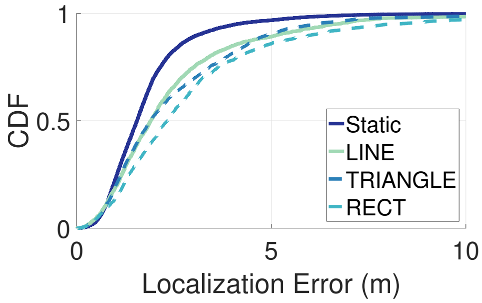


Figure 5.18: The three trajectories have similar localization error.

during such turning events (3$\times$ at 80 %ile).

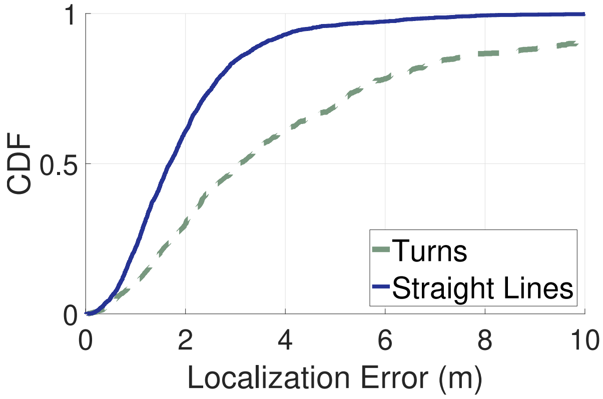


Figure 5.19: Localization error worsens during turns

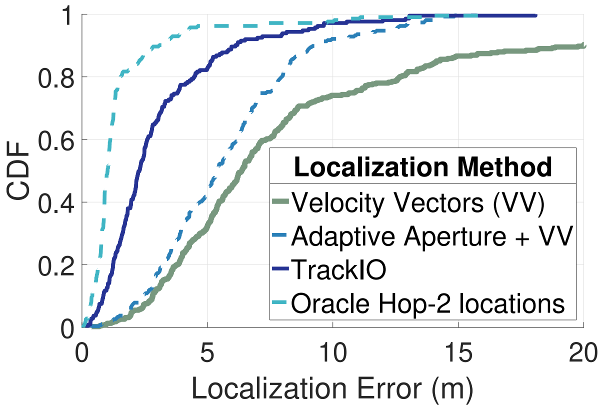


Figure 5.20: Benefit of spatial aperture vs. adaptive aperture

TrackIO tackle such cases through adaptively changing the aperture and spatial aperture offered by $\mathtt{hop}_2$ node positions to fix the erring $\mathtt{hop}_1$ nodes as described in §5.3. Fig. 5.20 shows the effectiveness of these approaches in our testbed and will be referred to in the following analysis.

Benefit of Adaptive Aperture. Fig. 5.21 shows a time series of localization error when a turning event occurs. The localization error increases sharply and remains large while the aperture slowly moves over this point in time. Our adaptive aperture dynamically resets historical measurements in case it detects turning events. It improves localization accuracy by a factor of $1.8\times$, and also reduces the time (*early recovery* $\approx$ 5 secs) it takes to stabilize the localization performance.

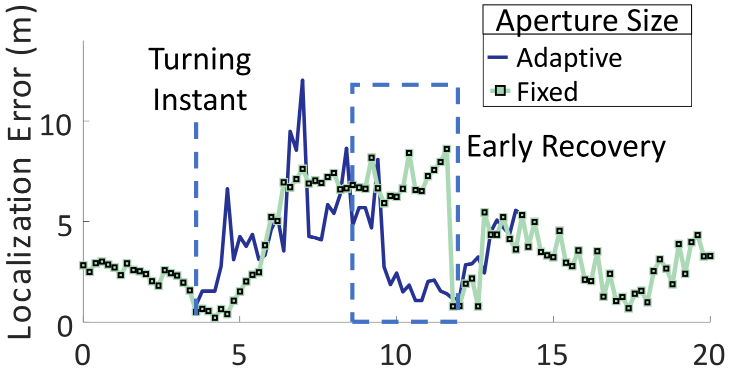


Figure 5.21: Adaptive aperture recovers from the effects of a turn earlier than fixed aperture.

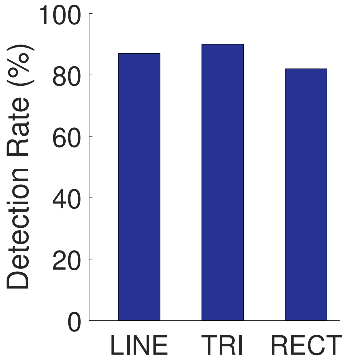


Figure 5.22: TrackIO correctly detects more than 80% of turns.

Benefit of spatial aperture from $\texttt{hop}_2$ nodes. Opportunistic re-fixing of a turning $\texttt{hop}_1$ node, T might be possible if there are at least three $\texttt{hop}_2$ nodes that do not depend on T for their localization. Fig. 5.20 shows the reduction in localization error when a fast moving and turning node is subjected to $\texttt{hop}_2$ guided re-fix. As an example, we use the entire $2\,$m/s run shown in Fig. 5.17 *including the turns* at either ends. The velocity-vector based localization performs poorly as seen from Fig. 5.20. We use 3 other static $\texttt{hop}_1$ nodes to localize 3 static $\texttt{hop}_2$ nodes and then use their locations to solve for the location of the running node, T. We observe about $2.8\times$ improvement in the median localization accuracy after re-fixing. This re-fixing accuracy is affected by two factors: (a) ranging error, and, (b) imprecise location of the $\texttt{hop}_2$ nodes. We can eliminate the effect of imprecise locations and hypothetically study just the ranging error effect by assuming ground truth $\texttt{hop}_2$ node locations are known—as if given by an Oracle. Fig. 5.20 shows this error to be within $2m$ at the 75%ile. While extremely promising, the re-fix approach may not always be available depending on the current topology. In comparison, the adaptive aperture technique is always available for any $\texttt{hop}_1$ node. Fig. 5.20 puts both these approaches (adaptive aperture and re-fix) into perspective.

Effect of drone trajectory and velocity. Fig. 5.23 shows the localization precision of two different trajectories, STRAIGHT and WAVY, at two different drone speeds. In general, geometric diversity of measurements helps obtain better localization. Therefore, the faster the drone moves, the better is the localization. Similarly, a WAVY pattern of drone movement also helps in obtaining better localization even for lower speeds. We therefore fly the drone in a WAVY pattern for this evaluation.

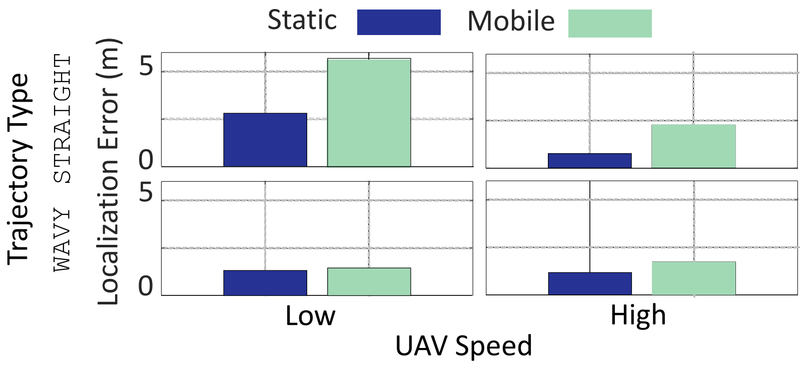


Figure 5.23: A WAVY trajectory provides better localization accuracy due to increase in spatial diversity.

5.5.2 Hop$_2$ Localization Performance

Effect of hop$_1$ mobility. We perform instantaneous localization[3] of hop$_2$ nodes based on 3 hop$_1$ nodes. By selecting which 3 nodes to use for this purpose, we obtain a combination of static and mobile hop$_1$ nodes—ranging from all static to all mobile. Fig. 5.24 shows the impact on localization errors as we allow an increasing number of hop$_1$ nodes to be mobile. These results show the error-span between using only static hop$_1$ nodes (1.83 m) to using only mobile hop$_1$ nodes (2.84 m). If multiple hop$_1$ nodes are available, choice of anchors influences hop$_2$ localization error (Fig. 5.25). Due to instantaneous localization of hop$_2$ nodes, the accuracies for both static and mobile hop$_2$ nodes are similar (see Fig. 5.15).

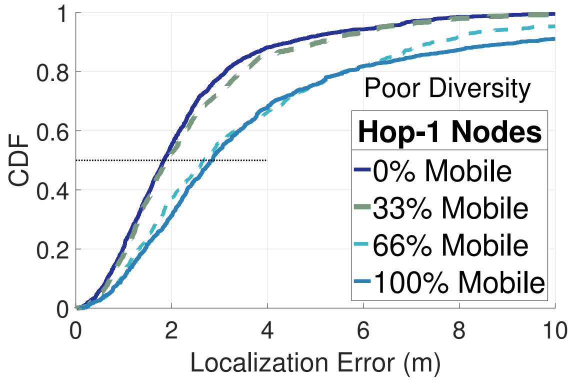


Figure 5.24: Localization error for a mobile hop$_2$ node with increasing number of mobile hop$_1$ nodes.

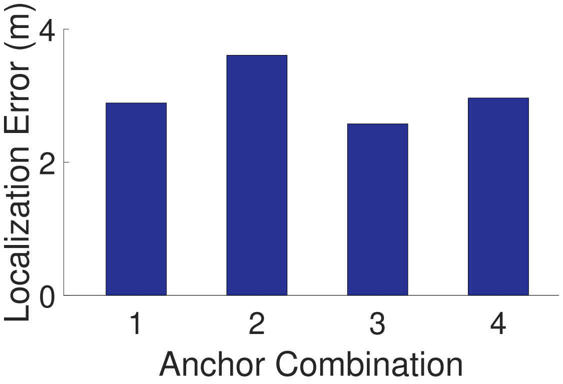


Figure 5.25: Selection of anchors influences localization error.

Effect of hop$_1$ diversity. The relative locations of hop$_1$ nodes also affect the localization accuracy. We consider random static snapshots of hop$_1$ node locations moving on the

[3] Instantaneous localization does *not* depend on UAV's synthetic aperture created in time.

three trajectories (LINE, TRIANGLE, and RECT) and further localize $\texttt{hop}_2$ nodes. Figure 5.26 shows a long tail indicating that some $\texttt{hop}_1$ position combinations perform poorly. A further analysis of such failing combinations reveals that $\texttt{hop}_1$ nodes are nearly collinear[4] in such cases, causing very high dilution of precision [69, 70, 71] (Fig. 5.27). We expect such situations to be minimal and short lived in real-life.

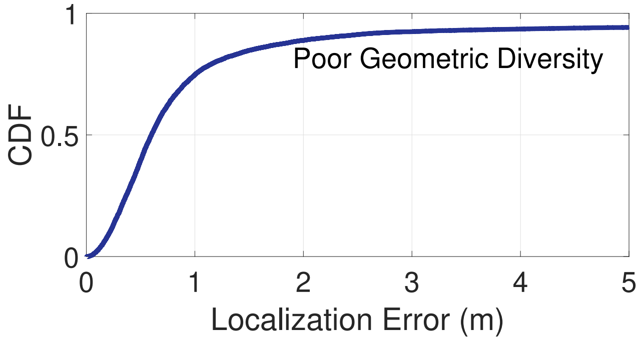


Figure 5.26: $\texttt{hop}_2$ localization error as a function of various $\texttt{hop}_1$ locations. Localization fails about 10% times.

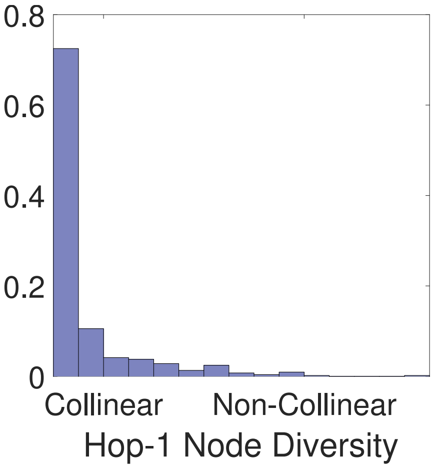


Figure 5.27: Most failure cases are caused by highly collinear $\texttt{hop}_1$ nodes.

5.5.3 UAV Localization

Note that the final localization accuracy of the mobile nodes is tied to absolute location fixes for the UAV. We study the impact of such accuracy as a function of the different modalities we can localize the UAV through (viz, COTS GPS receiver [72], UAV's GPS with sensor fusion [59], UWB based). Fig. 5.28 shows the error in each of these modalities compared to laser ranger based groundtruth (accurate to 1mm). Fig. 5.29 shows the improvement in localization error of a static indoor node (1 m using UWB versus 2+ m using GPS) when UWB based drone localization is used. Note that the UAV's large trajectory and the low vertical diversity in ground-UWB anchors degrades the drone's location accuracy. Yet, in GPS challenged situations, such as in a dense urban space, UWB based drone localization will remain valuable.

[4] We define collinearity as the ratio of the height and the base of the triangle formed by the three $\texttt{hop}_1$ nodes.

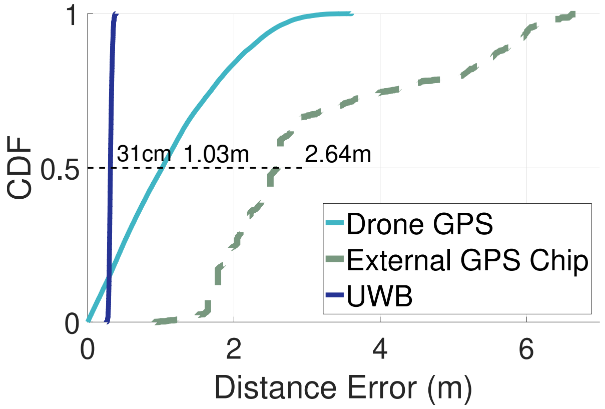


Figure 5.28: Range error between two fixed locations on the ground for different modalities.

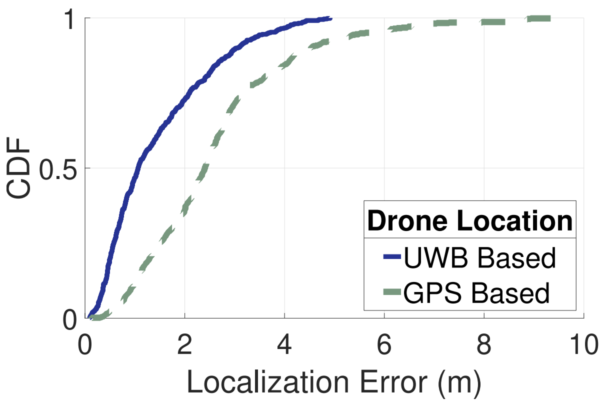


Figure 5.29: Effect of UAV localization modalities on static hop$_1$ localization accuracy.

5.5.4 TrackIO End-to-End Latency

We present TrackIO's end-to-end latency for different topologies in Figure 5.30. Each topology shown assumes an additional 4 hop$_1$ nodes on the ground for UAV localization. Thus, topology A consists of $4 + 6 = 10$ hop$_1$, and 3 hop$_2$ nodes. With even 20 nodes TrackIO leaves room for real-time operations.

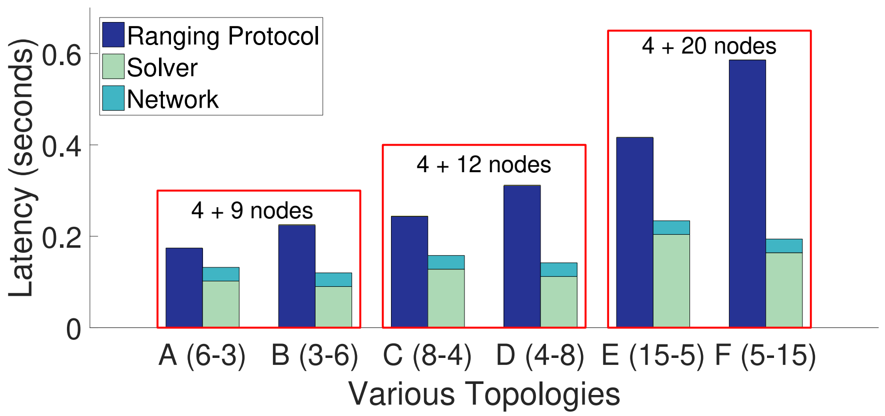


Figure 5.30: Protocol latency as a function of the network topology

5.5.5 Adding IMU: A What-If Analysis

Adding an IMU to our implementation might improve TrackIO's performance due to availability of another estimation for velocity and direction. We show in Fig. 5.31 and Fig 5.32 that while improvement in performance are possible when accurate direction and velocity information is available, presence of small errors in those estimates substantially reduce the gain over TrackIO. We leave more sophisticated IMU-based implementation to future work.

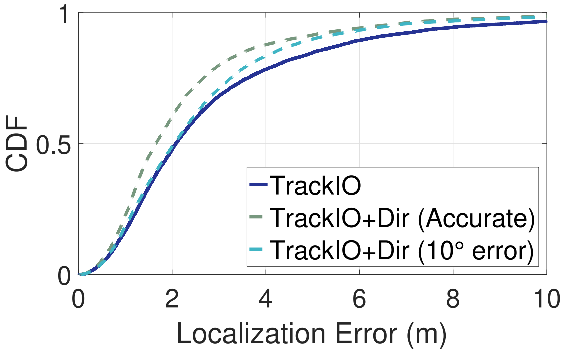


Figure 5.31: Improvement using direction from IMU

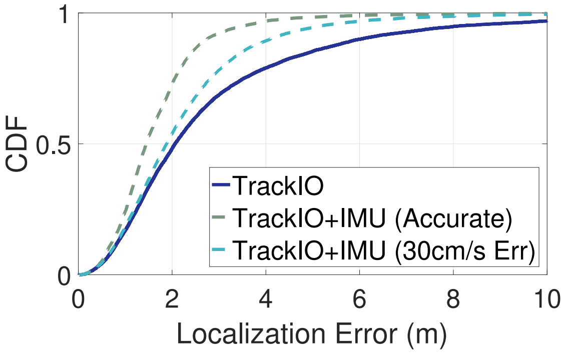


Figure 5.32: Improvement using velocity from IMU

5.6 RELATED WORK

Indoor localization. A significant amount of work exists in indoor localization [73, 74, 75, 76, 77, 78, 79], but most of it relies on indoor infrastructure and fingerprinting. Both of these are not available in our target application. Techniques that use commodity WiFi [80, 81] rely on the difference in subcarrier phases. However, subcarrier phase wraps after a short distance ($7.5 - 15\,\mathrm{m}$) rendering them unsuitable in our application. Localizing from outside a building could be performed with RF sensing [82, 83], however, we require a system that would be robust to changing multipath, fast human mobility, and would penetrate deep into a building. Use of inertial sensors (IMU) for tracking human motion [84, 79, 85] has been extensively studied. However, most of these systems suffer from drifts and saturation introduced by the IMU sensors [86, 1, 87]. In contrast to IMU-based tracking, the ranging approach we take in TrackIO is not based on dead-reckoning and instead provides instantaneous location.

UWB based localization. UWB radios are increasingly being used for localization solutions in a variety of applications from positioning [88, 89], to tracking industrial objects [90, 91], to sports analytics [1]. UWB is particularly resistant to indoor multipath [57, 92, 93] due to its $1ns$ time resolution ($1GHz$ bandwidth). This makes it suitable for use in indoor spaces where multipath can be rampant. Different UWB platforms are commercially available today [6, 94, 95], and we chose Decawave Trek1000 UWB platform for its superior performance [57, 96]. Most of these works assume some static UWB anchors. In our application however, we have no pre-deployed anchors, but create a synthetic aperture over time by flying an anchor on a UAV. Some recent works [97, 98] have explored use of the multipath profile as virtual anchors localize using a single UWB device. However, they assume the location of all strong reflectors are known, making it prone to issues when multipath could change, due to moving people or objects. In contrast, our technique does not depend on the knowledge of the floor-plan, and is robust to changing multipath profile.

Localization of UAV. UAV localization has been extensively studied and approaches range from using a single GPS [59, 62], to using differential GPS [39], to using complex motion models based on the drone's IMU data. UAV localization using UWB has been proposed in [99, 100]. We incorporate UAV localization into TrackIO protocol. Authors of [101, 102] also consider UAVs as a vehicle for fire-fighting, though they do not discuss the outdoor-indoor localization problem.

5.7 CONCLUSION

Indoor localization without any support from the building's infrastructure is a challenging yet important problem. Particularly of importance to first responders, continuous real-time tracking can be a life saver in many everyday situations. TrackIO uses a UAV to create the missing infrastructure *outside* the building and performs continuous ranging with indoor nodes. Through numerous algorithmic, architectural, and engineering modifications to trilateration and ranging protocols we obtain promising results localizing mobile indoor nodes accurate to about $2m$ from twenty meters outside the building. We believe TrackIO is a promising first step in active localization from outside the building. While TrackIO provides a fully working system where none exists today, we plan to continue to explore avenues to further improve accuracy, resilience, and redundancy.

CHAPTER 6: FUTURE WORK AND CONCLUSION

This work focuses on using wireless signals for applications that go beyond data communication. We have demonstrated the use of wireless time-of-flight for distance measurements and hence localization. We track a fast moving ball using angle-of-arrival together with ranging, and localize players in a constantly moving topology where some players are dynamically designated as anchors based on their geometry. We are now interested in the question: Is it possible to measure fine movements using localization techniques? We plan to continue this exploration through fine-grained CIR based analysis, potentially enabling virtual reality applications such as tracking the tilting of one's head. Figure 6.1 shows an example system that tracks head movement to around 5mm precision.

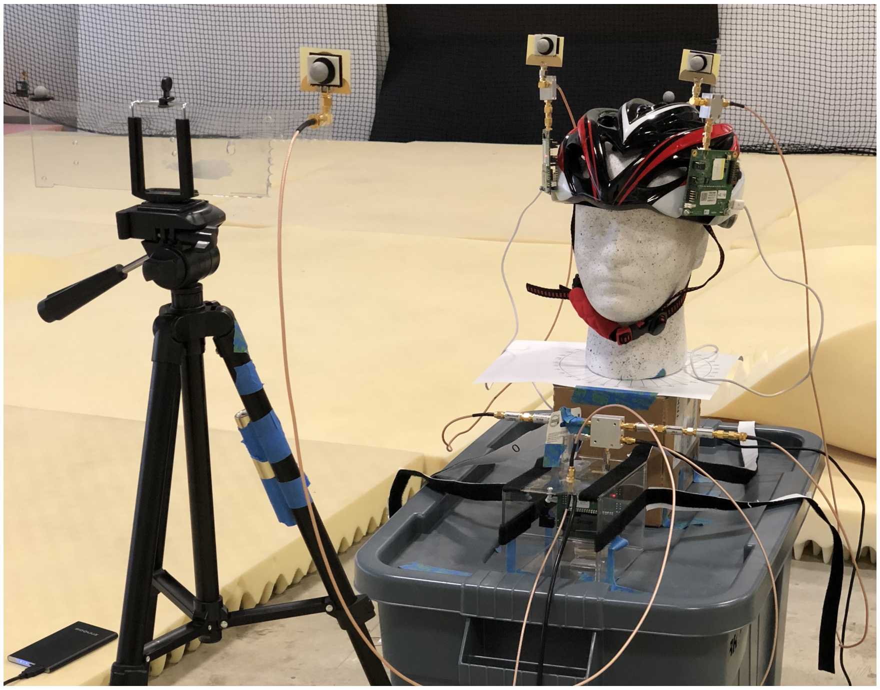


Figure 6.1: Wireless signals can enable precise motion tracking for applications in augmented and virtual reality.

In the material sensing space, we wish to make advances in using reflections for material identification, in addition to the refractions we used in LiquID. Reflections also provide a similarly rich set of information through attenuation, phase shift at the boundary, and polarization of RF signals. When using reflections, both the transmitter and the receiver can be on the same side of the object instead of being on either sides. This setup proves to be practical in a variety of situations where only one side of an object is accessible. For example, soil moisture detection can be performed by a hand-held device above the soil surface, self driving cars can distinguish between a tree and a person, a construction inspector can check the quality of building materials, and so on. We plan to explore if such a vision is feasible using wireless signals and have created a preliminary setup (Figure 6.2) that can accurately detect soil moisture levels.

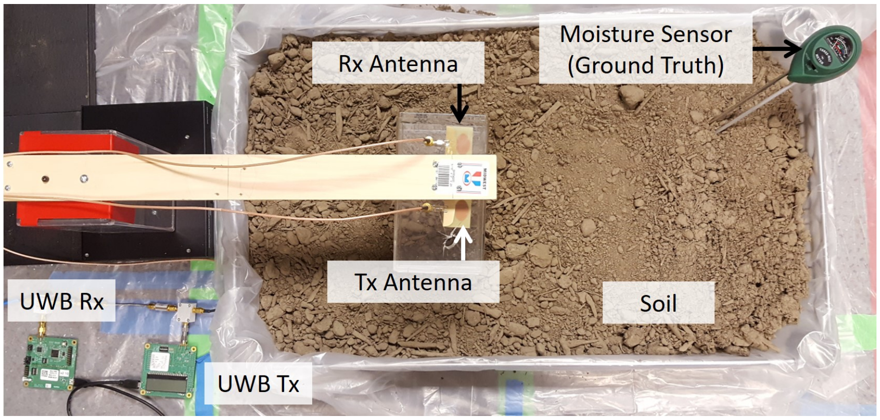


Figure 6.2: Wireless signals reflecting off the soil surface can be analyzed to measure soil moisture content.

In conclusion, we have shown the feasibility of extracting information imbibed into the wireless signals as they traverse our environments. In particular, we have shown how it is possible to identify materials and localize objects or people using wireless signals. We believe this capability can be extended to various sensing applications and we expect wireless sensing to soon occupy a center-stage akin to wireless communication.

REFERENCES

[1] M. Gowda, A. Dhekne, S. Shen, R. R. Choudhury, L. Yang, S. Golwalkar, and A. Essanian, “Bringing IoT to sports analytics,” in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. USENIX Association, pp. 499–513.

[2] A. Dhekne, U. J. Ravaioli, and R. R. Choudhury, “Localizing mobile topologies,” *under submission HotMobile*, 2019.

[3] A. Dhekne, M. Gowda, Y. Zhao, H. Hassanieh, and R. R. Choudhury, “Liquid: A wireless liquid identifier,” *Mobisys*, 2018.

[4] A. Hind, “Agilent 101: An introduction to optical spectroscopy,” 2013.

[5] A. A. Note, “Agilent basics of measuring the dielectric properties of materials,” *Agilent Literature Number*, 2006.

[6] DecaWave, “Decawave,” http://www.decawave.com/.

[7] B. Levitas, J. Matuzas, G. Viswanath, V. Basalingappa, and V. Venkoparao, “Uwb based oil quality detection,” in *Ultra-Wideband (ICUWB), 2011 IEEE International Conference on*. IEEE, 2011, pp. 220–224.

[8] P. Jaturatussanai, M. Chamchoy, and S. Promwong, “Characteristics of uwb propagation through building materials,” in *Communications and Information Technology, 2005. ISCIT 2005. IEEE International Symposium on*, vol. 2. IEEE, 2005, pp. 987–990.

[9] R. R. Selmic, A. Mitra, S. Challa, and N. Simicevic, “Ultra-wideband signal propagation experiments in liquid media,” *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 1, pp. 215–220, 2010.

[10] J. Wang, J. Xiong, X. Chen, H. Jiang, R. K. Balan, and D. Fang, “Tagscan: Simultaneous target imaging and material identification with commodity rfid devices,” in *Proc. ACM MobiCom*, 2017, pp. 1–14.

[11] A. R. Von Hippel, *Dielectrics and waves*. Wiley, 1954.

[12] “Reflection and refraction,” http://electron6.phys.utk.edu/optics421/modules/m1/reflection_and_refraction.htm.

[13] R. P. Feynman, R. B. Leighton, and M. Sands, *The Feynman lectures on physics, Vol. I: The new millennium edition: mainly mechanics, radiation, and heat*. Basic books, 2011, vol. 1.

[14] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar, and H. S. Al-Khalifa, “Ultra wideband indoor positioning technologies: Analysis and recent advances,” *Sensors*, vol. 16, no. 5, p. 707, 2016.

[15] "National instruments ni-tclk technology for timing and synchronization of modular instruments," http://www.ni.com/tutorial/3675/en/. [Online]. Available: http://www.ni.com/tutorial/3675/en/

[16] K. Blattenberger, "Dielectric Constant, Strength, and Loss Tangent." [Online]. Available: http://www.rfcafe.com/references/electrical/dielectric-constants-strengths.htm

[17] "N5242a PNA-X Microwave Network Analyzer," https://literature.cdn.keysight.com/litweb/pdf/N5242-90007.pdf?id=1118335. [Online]. Available: https://literature.cdn.keysight.com/litweb/pdf/N5242-90007.pdf?id=1118335

[18] "HP85070A Dielectric Probe Kit," https://literature.cdn.keysight.com/litweb/pdf/85070-90001.pdf?id=1326230. [Online]. Available: https://literature.cdn.keysight.com/litweb/pdf/85070-90001.pdf?id=1326230

[19] U. Kaatze, "Complex permittivity of water as a function of frequency and temperature," *Journal of Chemical and Engineering Data*, vol. 34, no. 4, pp. 371–374, 1989.

[20] C. Anderson, "Determining the complex permittivity of materials with the waveguide-cutoff method." Ph.D. dissertation, 2006.

[21] V. Komarov, S. Wang, and J. Tang, "Permittivity and measurements," *Encyclopedia of RF and microwave engineering*, 2005.

[22] A. P. Gregory and R. Clarke, *Tables of the complex permittivity of dielectric reference liquids at frequencies up to 5 GHz*. National Physical Laboratory Teddington, 2001.

[23] J. F. James, R. S. Sternberg, and S. A. Rice, "The design of optical spectrometers," *Physics Today*, vol. 23, p. 55, 1970.

[24] T. Rahman, A. T. Adams, P. Schein, A. Jain, D. Erickson, and T. Choudhury, "Nutrilyzer: A mobile system for characterizing liquid food with photoacoustic effect," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 2016, pp. 123–136.

[25] M. N. Afsar, J. R. Birch, R. Clarke, and G. Chantry, "The measurement of the properties of materials," *Proceedings of the IEEE*, vol. 74, no. 1, pp. 183–199, 1986.

[26] C. C. Courtney, "Time-domain measurement of the electromagnetic properties of materials," *IEEE Transactions on Microwave Theory and Techniques*, vol. 46, no. 5, pp. 517–522, 1998.

[27] H. Yue, K. L. Virga, and J. L. Prince, "Dielectric constant and loss tangent measurement using a stripline fixture," *IEEE Transactions on Components, Packaging, and Manufacturing Technology: Part B*, vol. 21, no. 4, pp. 441–446, 1998.

[28] "6 techniques for measuring dielectric properties," https://www.degruyter.com/downloadpdf/books/9783110455403/9783110455403-007/9783110455403-007.pdf.

[29] J. Baker-Jarvis, E. J. Vanzura, and W. A. Kissick, "Improved technique for determining complex permittivity with the transmission/reflection method," *IEEE Transactions on microwave theory and techniques*, vol. 38, no. 8, pp. 1096–1103, 1990.

[30] G. Wang, C. Qian, J. Han, W. Xi, H. Ding, Z. Jiang, and J. Zhao, "Verifiable smart packaging with passive rfid," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 156–166.

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

[32] Y. Ma, X. Hui, and E. C. Kan, "3d real-time indoor localization via broadband nonlinear backscatter in passive devices with centimeter precision," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 2016, pp. 216–229.

[33] T. Wei and X. Zhang, "Gyro in the air: tracking 3d orientation of batteryless internet-of-things," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 2016, pp. 55–68.

[34] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3d tracking via body radio reflections." in *NSDI*, 2014.

[35] B. Campbell, P. Dutta, B. Kempke, Y.-S. Kuo, and P. Pannuto, "Decawave: Exploring state of the art commercial localization," *Ann Arbor*, vol. 1001, p. 48109, 2015.

[36] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "Fingerio: Using active sonar for fine-grained finger tracking," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 2016, pp. 1515–1525.

[37] W. Mao, J. He, and L. Qiu, "Cat: high-precision acoustic motion tracking," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 2016, pp. 69–81.

[38] S. Yun, Y.-C. Chen, H. Zheng, L. Qiu, and W. Mao, "Strata: Fine-grained acoustic-based device-free tracking," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2017, pp. 15–28.

[39] M. Gowda, J. Manweiler, A. Dhekne, R. R. Choudhury, and J. D. Weisz, "Tracking drone orientation with multiple gps receivers," in *Mobicom*. ACM, 2016.

[40] "DW1000 IEEE802.15.4-2011 UWB Transceiver Datasheet," https://www.decawave.com/products/dw1000. [Online]. Available: https://www.decawave.com/products/dw1000

[41] "AccuraUWB Flex Series," http://cdn.taoglas.com/datasheets/FXUWB20.01.0100C.pdf. [Online]. Available: http://cdn.taoglas.com/datasheets/FXUWB20.01.0100C.pdf

[42] J. A. Gutierrez, E. H. Callaway, and R. Barrett, *IEEE 802.15.4 Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensor Networks*. IEEE Standards Office, 2003.

[43] B. TAO, "Understand the mobile graphics processing unit," http://www.embedded-computing.com/embedded-computing-design/understand-the-mobile-graphics-processing-unit. [Online]. Available: http://www.embedded-computing.com/embedded-computing-design/understand-the-mobile-graphics-processing-unit

[44] Y. Jiang and V. C. M. Leung, "An asymmetric double sided two-way ranging for crystal offset," in *ISSSE*, July 2007, pp. 525–528.

[45] M. McLaughlin and B. Verso, "Asymmetric double-sided two-way ranging in an ultra-wideband communication system," Feb. 14 2016.

[46] P. Massatt and K. Rudnick, "Geometric formulas for dilution of precision calculations," *Navigation*, vol. 37, no. 4, 1990.

[47] I. Guvenc, C.-C. Chong, and F. Watanabe, "Nlos identification and mitigation for uwb localization systems," in *WCNC*, 2007.

[48] DecaWave, "DW1000 User Manual," 2016.

[49] Vicon, "Vicon," 2018, https://www.vicon.com/.

[50] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.

[51] R. F. Fahy, P. R. LeBlanc, and J. L. Molis, "Firefighter fatalities in the united states - 2017," *NFPA*, 2018.

[52] 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*. ACM, 2012, pp. 197–210.

[53] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. Ieee, 2000, pp. 775–784.

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

[55] L. Ni, Y. Wang, H. Tang, Z. Yin, and Y. Shen, "Accurate localization using lte signaling data," in *2017 IEEE International Conference on Computer and Information Technology (CIT)*, Aug 2017, pp. 268–273.

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

[57] A. R. J. Ruiz and F. S. Granja, "Comparing ubisense, bespoon, and decawave uwb location systems: Indoor performance analysis," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 8, pp. 2106–2117, Aug 2017.

[58] J. Palacios, P. Casari, and J. Widmer, "Jade: Zero-knowledge device localization and environment mapping for millimeter wave systems," in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 2017, pp. 1–9.

[59] DJI, "DJI Phantom 4 Specs."

[60] F. van Diggelen and P. Enge, "The worlds first gps mooc and worldwide laboratory using smartphones," in *Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, 2015, pp. 361–369.

[61] W. J. T. C. N. T. Hughes and E. Team, "Global positioning system (gps) standard positioning service (sps) performance analysis report," *WAAS FAA*, no. 101, 2018.

[62] DJI, "Mavic pro specs," https://www.dji.com/mavic/info#specs.

[63] Pix4D, "Do rtk/ppk drones give you better results than using gcps?" https://pix4d.com/rtk-ppk-drones-gcp-comparison/.

[64] W. Hedgecock, M. Maroti, A. Ledeczi, P. Volgyesi, and R. Banalagay, "Accurate real-time relative localization using single-frequency gps," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. ACM, 2014, pp. 206–220.

[65] W. Hedgecock, M. Maroti, J. Sallai, P. Volgyesi, and A. Ledeczi, "Regtrack: a differential relative gps tracking solution," in *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 2013, pp. 475–476.

[66] B. Parkinson and P. Enge, "Differential gps," in *Global Positioning System: Theory and Applications*. American Institute of Aeronautics and Astronautics, 1996, pp. 3–50.

[67] A. Irish, J. Isaacs, D. Iland, J. Hespanha, E. Belding, and U. Madhow, "Shadowmaps, the urban phone tracking system," in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014, pp. 283–286.

[68] D. Developer, "Mobile sdk," https://developer.dji.com/mobile-sdk/documentation/introduction/index.html.

[69] R. Yarlagadda, I. Ali, N. Al-Dhahir, and J. Hershey, "Gps gdop metric," *IEE Proceedings-radar, sonar and navigation*, vol. 147, no. 5, pp. 259–264, 2000.

[70] E. Swanson, "Geometric dilution of precision," *Navigation*, vol. 25, no. 4, pp. 425–429, 1978.

[71] P. Massatt and K. Rudnick, "Geometric formulas for dilution of precision calculations," *Navigation*, vol. 37, no. 4, pp. 379–391, 1990.

[72] "Adafruit ultimate gps breakout," https://www.adafruit.com/product/746.

[73] 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)*, 2016, pp. 165–178.

[74] S. Kumar, S. Gil, D. Katabi, and D. Rus, "Accurate indoor localization with zero start-up cost," in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014, pp. 483–494.

[75] P. Bahl and V. N. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2. Ieee, 2000, pp. 775–784.

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

[77] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. ACM, 2005, pp. 205–218.

[78] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proceedings of the 18th annual international conference on Mobile computing and networking*. ACM, 2012, pp. 269–280.

[79] 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*. ACM, 2012, pp. 293–304.

[80] K. R. Joshi, D. Bharadia, M. Kotaru, and S. Katti, "Wideo: Fine-grained device-free motion tracing using rf backscatter." in *NSDI*, 2015, pp. 189–204.

[81] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," in *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 269–282.

[82] F. Adib, C.-Y. Hsu, H. Mao, D. Katabi, and F. Durand, "Capturing the human figure through a wall," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 219, 2015.

[83] F. Adib, Z. Kabelac, and D. Katabi, "Multi-person localization via rf body reflections." in *NSDI*, 2015, pp. 279–292.

[84] 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*. ACM, 2012, pp. 197–210.

[85] A. R. J. Ruiz, F. S. Granja, J. C. P. Honorato, and J. I. G. Rosas, "Accurate pedestrian indoor navigation by tightly coupling foot-mounted imu and rfid measurements," *IEEE Transactions on Instrumentation and measurement*, vol. 61, no. 1, pp. 178–189, 2012.

[86] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *arXiv preprint arXiv:1704.06053*, 2017.

[87] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-markie: Indoor pathway mapping made easy," in *Proceedings of the 10th USENIX conference on Networked Systems Design and Implementation*. USENIX Association, 2013, pp. 85–98.

[88] 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 Trans. Sen. Netw.*, vol. 14, no. 2, pp. 11:1–11:29, June 2018. [Online]. Available: http://doi.acm.org/10.1145/3185752

[89] A. Prorok, A. Arfire, A. Bahr, J. R. Farserotu, and A. Martinoli, "Indoor navigation research with the khepera iii mobile robot: An experimental baseline with a case-study on ultra-wideband positioning," in *2010 International Conference on Indoor Positioning and Indoor Navigation*, Sept 2010, pp. 1–9.

[90] S. Spieker and C. Rohrig, "Localization of pallets in warehouses using wireless sensor networks," in *2008 16th Mediterranean Conference on Control and Automation*, June 2008, pp. 1833–1838.

[91] J.-A. Fernandez-Madrigal, E. Cruz-Martin, J. Gonzalez, C. Galindo, and J.-L. Blanco, "Application of uwb and gps technologies for vehicle localization in combined indoor-outdoor environments." in *ISSPA*, 2007, pp. 1–4.

[92] S. Sczyslo, J. Schroeder, S. Galler, and T. Kaiser, "Hybrid localization using uwb and inertial sensors," in *2008 IEEE International Conference on Ultra-Wideband*, vol. 3, Sept 2008, pp. 89–92.

[93] A. Prorok, P. Tom, and A. Martinoli, "Accommodation of nlos for ultra-wideband tdoa localization in single- and multi-robot systems," in *2011 International Conference on Indoor Positioning and Indoor Navigation*, Sept 2011, pp. 1–9.

[94] BeSpoon, “Precise location rtls,” http://bespoon.com/technology/precise-location-rtls/.

[95] Ubisense, “Ubisense solutions,” http://www.ubisense.net/ubisense-solutions.

[96] A. R. Jiménez and F. Seco, “Comparing decawave and bespoon uwb location systems: Indoor/outdoor performance analysis.” in *IPIN*, 2016, pp. 1–8.

[97] S. Van de Velde and H. Steendam, “Cupid algorithm for cooperative indoor multipath-aided localization,” in *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*. IEEE, 2012, pp. 1–6.

[98] J. Kulmer, S. Hinteregger, B. Großwindhager, M. Rath, M. S. Bakr, E. Leitinger, and K. Witrisal, “Using decawave uwb transceivers for high-accuracy multipath-assisted indoor positioning,” in *Communications Workshops (ICC Workshops), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1239–1245.

[99] 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*. ACM, 2015, pp. 16–20.

[100] A. Benini, A. Mancini, and S. Longhi, “An imu/uwb/vision-based extended kalman filter for mini-uav localization in indoor environment using 802.15. 4a wireless sensor network,” *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 461–476, 2013.

[101] C.-E. Hrabia, A. Hessler, Y. Xu, J. Brehmer, and S. Albayrak, “Efffeu project: Efficient operation of unmanned aerial vehicles for industrial fire fighters,” in *Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, ser. DroNet’18. New York, NY, USA: ACM, 2018. [Online]. Available: http://doi.acm.org/10.1145/3213526.3213533 pp. 33–38.

[102] E. DUEWEL and S. STODDARD, “After dark, drone is ’best friend a firefighter could have’,” https://www.usnews.com/news/best-states/oregon/articles/2018-08-11/after-dark-drone-is-best-friend-a-firefighter-could-have.