UTrack3D: 3D Tracking Using Ultra-wideband (UWB) Radios

Yifeng Cao
ycao361@gatech.edu
Georgia Institute of Technology
USA

Ashutosh Dhekne
dhekne@gatech.edu
Georgia Institute of Technology
USA

Mostafa Ammar
ammar@cc.gatech.edu
Georgia Institute of Technology
USA

ABSTRACT

Recording 3D movements of a user's hand, robotic arms, or an object, even in a small confined space, has several applications in AR/VR, robotics, movement science, and 3D modeling and rendering. Existing camera-based tracking systems, though extremely accurate, are quite expensive and suffer from issues of occlusion and face difficulties when operating in extremely dark or extremely bright environments. We contend that trading-off a bit of accuracy while reducing costs and enabling more flexible operating environment might be worth exploring. This paper presents UTrack3D, a table-top setup that tracks the movements of an object in 3D space using embedded low-cost ultra-wideband (UWB) radios. The core idea is to continuously track the changes in phase as captured from UWB signal's channel impulse response (CIR) derived from the UWB messages received at a set of dual-antenna UWB receivers. Each of our custom dual-antenna receivers captures the UWB signal from two corners of a cuboid allowing us to perform relative phase measurements. The main challenges in the solution are caused by a location-dependant large variation in the signal amplitudes and corruption of the CIR due to multipath. UTrack3D tackles these challenges via a signal processing pipeline fusing a forward localization process which tracks the object's location using UWB CIR phase, and a posterior location check process, which validates the estimated location. UTrack3D is implemented on commercial-off-the-shelf (COTS) UWB chips, and provides a 90th percentile accuracy of $9\,mm$ in a table-top 3D region ($1.5m \times 0.8m \times 0.8m$). We evaluate the effects of additional UWB receivers, effect of different movement speeds, and effect of small-scale signal blocking using different materials. We expect UTrack3D to allow researchers a rich new environment for further advancing UWB-based 3D tracking.

CCS CONCEPTS

• **Computer systems organization** → **Sensors and actuators**; • **Human-centered computing** → **Ubiquitous and mobile computing**.

KEYWORDS
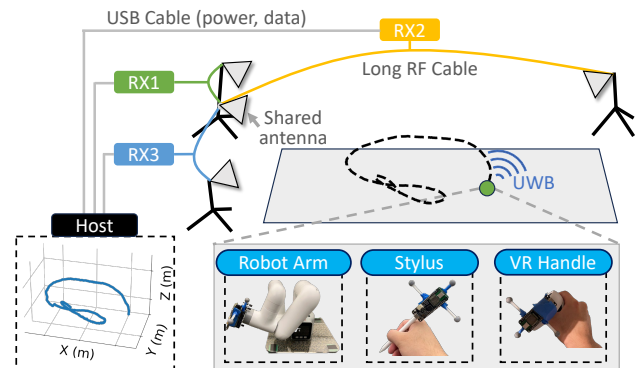
UWB-based 3D Tracking, UWB PDoA, UWB TDoA

Figure 1: Overview of UTrack3D setup with three RF-receivers each with a dedicated antenna, while also sharing 1 common antenna, and UTrack3D use-cases.

1 INTRODUCTION

Tracking 3D movement of hands, robotic arms, or 3D objects, even when confined to a small area such as over a work-bench or table-top is a fundamental primitive with applications spanning from robotics, to gaming, to AR/VR, to sculpting, to music. The core capability required in many of these applications appears to be quite simple: track the 3D movements of an object in a small confined 3D space. Today, stereoscopic cameras are able to perform such 3D tracking, and are actually far more capable in terms of coverage area and accuracy than what we need for a table-top solution. However, cameras are often constrained by occlusions, they need to detect the object clearly with enough contrast from the background, they work only in well-lit conditions, and require a substantial compute power for running computer vision algorithms [34, 39, 46]. While substantial work has occurred in each of these deficient areas, the fundamental issues remain. Infrared markers that are clearly visible to infrared cameras are used to solve some of the problems faced by visible spectrum cameras, however, such motion capture solutions can be expensive (in excess of $10,000). The costs are quite fundamental, since specialized IR cameras and specialized processing software is expensive to produce. In this paper, we ask: *Is it possible to have graceful degradation in accuracy as we lower the cost of 3D tracking using an alternative technology, in a small area?*

To explore an alternative, we must contrast with the very mechanism that most existing motion capture systems use. At a high-level optical motion capture systems (both visible spectrum and infrared (IR) based) observe *reflections* from the object being tracked. They must identify the object's reflections and ignore all others (noise, in a signal processing sense) which is an inherently complex and compute-intensive operation. Now, instead of the infrastructure observing reflections from the object of interest, if the object itself emits identifiable messages, it becomes far easier to eliminate noise and track just the object we are interested in. Certain optical motion capture systems use coded light emitted from LEDs installed on

the object [52] eliminating the identification problem associated with reflective motion capture. However, since they also use light signals, occlusions are frequent and the surface of the object has to accommodate the LEDs. In comparison, radio frequency wireless signals emanating from a radio embedded inside the object can pass through various materials, without modifying the external appearance of the object, and the signals can be easily identified and localized by RF receivers placed in the infrastructure. These properties are promising and might provide a viable alternative to the currently prevalent motion capture systems.

Others have made similar observations about camera-based systems. For example, [21] explored these same alternatives, and embedded ultra-wideband (UWB) and inertial measurement unit (IMU) chips inside a ball. This approach has also been used in FiFA football [47]. In a different context, our earlier work [15] embedded an ultra-wideband chip in a pen to track its motion. In [59], a pen is passively tracked using millimeter-wave radio. However, a solution to the general 3D motion tracking problem remains elusive. Without a 2D plane to guide the possible locations as in ITrackU [15], or the parabolic trajectory of the falling ball that generates strong priors in iBall [20], accurate wireless 3D localization must be performed with minimal constraints in a general context. Yet, the existence of prior art in this domain shows that RF-based 3D tracking is a desirable property in a variety of applications.

In this work, we are interested in developing a methodology for free-form 3D tracking using an ultra-wideband (UWB) RF transmitter at the object being tracked. We will explore the benefits of UWB over other RF technologies such as Wi-Fi and RFID when we discuss related work in Section 2. We envision a wide range of use-cases for our system, called UTrack3D, spanning from tracking a robotic arm for precise manipulation, to drawing virtually in the air using a VR stylus, as shown in Fig. 1.

Our earlier work, ITrackU [15], tracks a pen on a 2D surface using a multi-antenna system with a single RF-chain. When we first set out to develop the ideas of this project, we thought it would be a trivial extension of the ITrackU work by adding one more antenna in the *Z-axis*. However, extending the ideas that work in 2D to 3D turns out to be non-trivial. We faced severe dynamic range issues and significant issues around non-ideal antenna transmission pattern where signal from the tracked object would be substantially attenuated with respect to at least one of the antennas at almost all locations. We feel our observations in extending ITrackU to 3D are instructive and therefore, we will dwell on more details of our exploration in Section 4.1.

While our first attempt to extend ITrackU by adding just one more antenna failed, we took inspiration from the basic principles and investigated how it could be modified for a 3D tracking system. UTrack3D is the result of that exploration. The foundational idea is as follows: ultra-wideband based localization is limited by its bandwidth if we only rely on timing information and perform two-way ranging. However, tracking can benefit also from the recorded *phase information* which can be obtained at a much finer granularity. Of course, received phase is arbitrary for two devices not sharing a clock. To use phase information, we must create a system that can track relative phase between two pulses received in the same channel impulse response (CIR) inferred from a single wireless packet. In UTrack3D, we obtain such relative phase using

independent dual-antenna setups along each of the three axes (see Fig. 1). Meaning, we have three separate UWB receivers, each catering to one axis, which mitigates dynamic range issues. Antennas along each axis are separated by a substantial distance ($\sim 1.5\,m$ and $\sim 0.8\,m$), but connected via a long RF cable, thus obtaining two signal pulses, separated by a few nanoseconds. Being part of the same CIR, such pulses can be compared in phase.

Our contributions in UTrack3D are:

• We propose a **3D tracking system using only UWB,** operating in a small confined space, such as a table-top, without support from other sensors. We utilize custom-made multi-antenna UWB receivers which captures the 3D location of a UWB-embedded object at sub-centimeter-level accuracy.

• We develop a signal processing pipeline to effectively enable a consistent and robust tracking process, resilient to various practical issues including multipath, low SNR, and even signal blocking.

• We prototype UTrack3D with COTS UWB chips and demonstrate its feasibility in tracing hand-drawn 3D shapes, and tracking the motion of a robotic arm.

UTrack3D performs tracking using only RF-based tracking without help from other sensors such as inertial sensors. This is a conscious decision since it helps us explore the capabilities and limitations of UWB-phase based tracking, simplifies the UWB communication interface with the transmitter sending just a sequence number instead of streaming inertial data, and allows us to treat UWB-based 3D tracking as a foundational technology. Based on the constraints and liberties of specific applications, researchers in the future might use the techniques developed in this work along with other sensor technologies.

We evaluate UTrack3D in our lab environment with OptiTrack based motion capture solution for ground truth. We obtain $9\,mm$ accuracy at the 90th percentile for free-form writing in the air and tracing a 3D object in a confined $1.5m \times 0.8m \times 0.8m$ table-top area. We verify that UTrack3D does not drift over time, study the effect of adding more receivers in the infrastructure, study the effect of different speeds of 3D movement, and explore absolute localization possibilities. We evaluate UTrack3D's performance under occlusions created by different materials, and evaluate the latency of the real-time processing pipeline. Our evaluations are performed in a cluttered lab so as to experience real-world multipath.

2 RELATED WORK

The broad application space that requires precise 3D tracking has motivated various tracking solutions based on different modalities. The literature makes it clear that any one solution may not be suitable for all applications. In this section, we briefly review existing solutions starting with marker-based, then dwelling on marker free, acoustic, and finally radio-frequency based solutions.

In **marker-attached camera-based solutions**, special infrared (IR) reflecting markers are placed on the object or human subject being tracked in a distinct pattern allowing a set of specialized IR cameras to detect their location. Several commercial solutions exist [6–8] and are widely used in movie making, movement sciences, and robotic tracking. Multiple infrared light emitters cover a fixed volume of space being tracked. The IR light reflects from the markers on the object and these reflections are captured by IR cameras (usually collocated with the IR emitters). While localization

accuracy can be sub-millimeter, such motion capture systems are very expensive[1].

When the tracking occurs on a single plane, such as a white-board, commercial solutions such as Boxlight's Mimio platform have driven down the costs (to around $1000) by combining infrared and acoustics. The Mimio platform is useful for tracking a lecturer's whiteboard writing, but is not a general solution for 3D spaces. In comparison, UTrack3D achieves sub-centimeter-level tracking accuracy in 3D space with low-cost sensors.

Marker-free camera-based solutions are well-studied in the computer vision field. One can either use external RGB-depth camera [4, 5], stereo cameras [9] to directly capture the position change of an object in the FOV [23, 27, 28, 31–33, 41], or perform simultaneous localization and mapping (SLAM) to infer the object's own position using an on-object camera [12, 42, 43, 48, 53]. However, object tracking using image data requires a compute-heavy pipeline involving object detection, recognition, and geometric transformation. Yet, its tracking accuracy is limited by the image resolution and are sensitive to external factors, such as ambient light, viewing position, occlusion, etc. In comparison, UTrack3D actively tracks the tag on the object using the UWB CIR at the receivers, yields much more efficient bandwidth usage and results in a simpler data processing pipeline without the need for object detection and object recognition. UTrack3D is unaffected by different lighting conditions, and works even when encased in different non-metallic materials.

Acoustic-based tracking solutions: Acoustic signals have been explored as an alternative to camera-based solutions for its low-cost sensors [13, 38, 45, 55, 58, 65, 66, 68, 70]. For instance, Strata [66] leverages the phase change in the acoustic CIR to passively track an object with a $1.0cm$ 2D tracking error. ReflecTrack [70] achieves a median error of $28.4mm$ in the $0.6m \times 0.6m \times 0.6m$ space using frequency-modulated continuous wave (FMCW) signals in audible frequencies. However, acoustic signals have limited range (typically $< 1m$). Modern built-in speakers and microphones suppress the power of acoustic signals in the inaudible range, leading to rapid attenuation. In addition, hardware imperfection generates noise even if inaudible frequencies [51], and can be irritating to pets and children and captured by nearby microphones. Acoustic signals also attenuate significantly when passing through various packaging materials such as wood, wool, and glass, etc. It is worth noting, however, that since sound travels much slower than radio-frequency signals, in some applications where the limitations of sound-based tracking do not apply, acoustic tracking can be a more practical and cost effective solution.

RF-based tracking solutions: High-precision RF-based tracking is an active research area and has seen substantial work in recent times. Existing RF-based solutions use RFID [29, 54, 57, 64], millimeter-wave signals[50, 59], Wi-Fi signals [26, 30, 60, 61], or custom RF signals [35, 62]. Millimeter-wave techniques are based on reflections of signals and suffer from many of the limitations of markerless camera systems. The tracking approach for RF-based systems is based on received signal strength (RSS), time of flight, or phase information. RSS-based techniques only provide coarse-grained localization and tracking. Time of flight solutions are limited in accuracy by bandwidth of the signal and in that sense, Wi-Fi

based solutions perform poorly compared to mmWave and UWB-based solutions. When phase information is used, stability of the phase measurements and the signal wavelength are crucial. For example, RFID phase $90\%tile$ stability for Impinj R420 at $907MHz$ is about $8.6°$ [49], which translates to $7.9\,mm$ theoretical limit. In contrast, UWB signals at $4\,GHz$ provides $0.9\,mm$ theoretical resolution limit [15]. Tracking based on RFID can be fragile when faced with multipath, flexing of tags or orientation changes, as reported by [56]. Use of Wi-Fi CSI is a possible alternative [71]. However, given how important and prevalent Wi-Fi communication is today, it is difficult to find an empty channel to dedicate to tracking. Without a dedicated channel, Wi-Fi tracking will suffer from interference from other communication devices in the vicinity and will itself interfere. RF-based solutions are also significantly influenced by environmental multipath. UWB signals with their large bandwidths are more resistant to multipath providing a promising alternative [14, 16–19, 40, 63]. Prior work has shown the promise of UWB-based tracking by using non-standard UWB signals emitted by custom hardware [36, 37, 67]. However, the hardware design of these efforts does not comply with the current UWB standards: they require much larger bandwidth than the standards, and the signals are captured using specialized oscilloscopes. Those systems, while pioneering at the time, are far from being an IoT platform for practical everyday use. Our previous work, ITrackU [15], fuses UWB and inertial sensors to track the motion of a pen-like instrument on a 2D plane, providing $7mm$ accuracy at the $90\%tile$. This current work, instead, uses only UWB signals (without fusion with other sensors) to achieve $9mm$ error in *3D space*, opening up opportunities for a much wider application space in the real-world.

Overall, we believe compared to existing work in this space, UTrack3D significantly advances 3D motion tracking and will find uses in a large number of applications.

3 PRIMER ON UWB PHASE-BASED LOCALIZATION

Before delving into the design of UTrack3D, it is useful to first describe the fundamentals of how UWB can leverage physical-layer (PHY) features to track the motion of a tag. Note that *UWB does not use FMCW*, but rather uses wide-band short-time pulses which make it simple to obtain the channel impulse response (CIR) [25].

Localization with channel impulse response. Traditionally, UWB employs two-way ranging which exchanges messages between two devices and measures the time-of-flight (ToF) based on transmission and reception timestamps to calculate the distance. While it is widely used as an active ranging solution in indoor localization applications [16, 24], the $10cm$ ranging error is inadequate for tracking applications that require at least an order of magnitude better precision, e.g., digitizing the trajectory of a AR-wand in the air. This paper takes a different approach which tracks with the phase change of the UWB's carrier wave which is available in the form of complex channel impulse response from our hardware. The basic model of UTrack3D involves a single-antenna UWB transmitter (TX), and multiple multi-antenna UWB receivers (RX). The TX continuously sends UWB messages, which are captured by each UWB receiver after traveling in the air for time τ_i. Note that the in-air propagation time τ_i is directly correlated to the TX-RX distance d_i by $\tau_i = \frac{d_i}{c}$, which is the primary constraint we use to perform

---

[1]We have received several price-quotes all exceeding $10,000 and have actually deployed one such system for our ground truth.
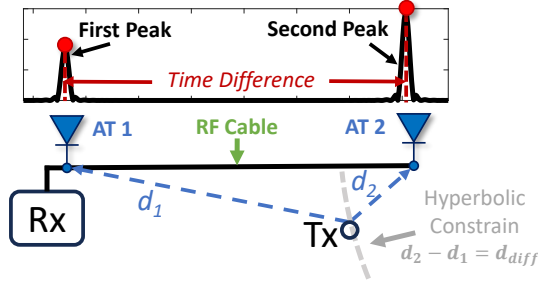
Figure 2: A 2-antenna receiver observes 2 peaks in CIR. The time difference of these two peaks build a hyperbolic constrain for localizing the transmitter.

localization. For ease of illustration, we temporarily neglect the multipath effect and assume the signal will only reach the antenna via the direct path. For a UWB signal with center frequency f_c, the received signal at receiver r_i is a copy of the transmitted signal $x(t)$ with a specific time delay τ and attenuation factor a:

$$y(t) = ae^{-j2\pi f\tau + \phi_o}x(t - \tau), \quad (1)$$

where ϕ_o is the phase offset varying with time and UWB receivers. With the assumption of linear channel, the input-output-irrelevant channel impulse response (CIR) between the TX and RX is modeled as:

$$h_i(t) = a_i e^{-j2\pi f\tau + \phi_o}\delta(t - \tau) = a_i e^{-j(2n\pi + \phi + \phi_o)}\delta(t - \tau), \quad (2)$$

where n is the number of phase wrappings. Obviously, the CIR is influenced by the location of the transmitter: the phase wrappings n and the fractional part ϕ are directly linked to the TX-RX distance with $d = (n + \frac{\phi}{2\pi})\lambda$. After analog-to-digital-conversion, the discretized CIR is a sequence of complex numbers which is extracted by correlating the received signal with known preamble. The phase observable from the CIR samples, ϕ^c, is actually the sum of in-air phase ϕ and phase offset ϕ_o ($\phi^c = \phi_o + \phi$), and does not include any information about the integer number of phase wraps. Next, we will introduce the method to cancel out the receiver-specific phase offset, and the integer phase wraps.

Phase offset cancellation with multiple antennas. The root cause of the receiver-specific phase offset is that a wireless transmitter cannot be phase-synchronized with a receiver since both run their own independent oscillators with their own drifts. To cancel the phase offset, we employ the same method as used in [15]. Rather than use a single-antenna receiver, we connect multiple antennas to a receiver, as shown in Fig. 2. These antennas are physically placed at different positions, connected with RF cables. Because all antennas in the same receiver share the same phase offset in one UWB transmission, we can differentiate the phases measured at each antenna to cancel phase offset. This way, the location of the UWB transmitter is inferred from the distance difference to each antenna. Mathematically, if we use 0-indexed antenna as the reference for a $k-$antenna receiver, the distance difference $\triangle d_i$ between the 0th antenna and the ith antenna, $i = [i..k]$ is calculated as:

$$d_i - d_0 = (n_i + \frac{\phi_i^c - \phi_0}{2\pi})\lambda - (n_0 + \frac{\phi_0^c - \phi_0}{2\pi})\lambda$$
$$= (n_i - n_0 + \frac{\phi_i^c - \phi_0^c}{2\pi})\lambda, \quad (3)$$

Every pair of antennas provides a hyperbola constraint to solve the location of the transmitter.

Integer phase-wrap cancellation. Obtaining the integer number of phase wraps is a hard problem in wireless communication, known as the *integer ambiguity problem* [22]. However, if the UWB packet rate is sufficiently fast, the phase change between two consecutive packets is within π. That way, the integer part can be simply dropped if we keep tracking the phase change between consecutive packets. More formally, the change of the distance difference in two adjacent UWB packets j and $j + 1$ can be written as

$$\triangle d_{i,j+1} - \triangle d_{i,j} = (\frac{\phi_{i,j+1}^c - \phi_{0,j+1}^c}{2\pi} - \frac{\phi_{i,j}^c - \phi_{0,j}^c}{2\pi})\lambda. \quad (4)$$

UTrack3D uses Equation (4) to track the object's 3D location with 3 antenna-pairs.

4 SYSTEM DESIGN

We now start with an exploration of the hardware setup followed by the system overview and then the system details.

4.1 Infrastructure Hardware Setup

4.1.1 Design considerations: As described in Section 3, one needs at least 3 pairs of air path differences to uniquely determine the location of the UWB transmitter. An intuitive implementation is using a single receiver with four antennas—one at the origin and three others at three extreme points along the three axes. This would be similar to ITrackU [15] that builds a receiver with 3 antennas to track a pen trajectory on a 2D surface. However, we observe the following issues of such a 3D tracking setup:

(i) Dynamic range issue: The amplitude of the impulses in CIR varies with the relative location of the TX to the antenna (distance and angle) and the radiation pattern of the antenna. Ideally, the received signal from each antenna should have comparable amplitude so the phases of all impulses can be clearly extracted. However, this is extremely difficult on a receiver with more than 3 antennas: the relative ratio between impulses dramatically change when the TX moves, and the weakest impulse is frequently overwhelmed by noise. Fig. 3 shows the observed CIR variation for a four-antenna receiver when we only move the TX by $30cm$. We observe that even this small movement can make the 3rd and the 4th impulse extremely weak, resulting in a tracking failure. Of course, moving in a different direction would weaken a different combination of impulses and get buried in noise.

(ii) Multipath issue: Apart from the direct line-of-sight (LOS) path, the transmitted signal can also reach the antenna after being reflected by other objects, resulting in a peak which comes a few nanoseconds later than the LOS signal. This is known as multipath. While the large bandwidth of UWB provides superior separability of these multipath peaks, an impulse selection algorithm needs to determine which peak corresponds to per-antenna LOS signal. This task is progressively harder with more antennas connected to the same receiver in presence of multipath. As shown in Fig. 3, the multipath of previous signals can confuse the peak selection algorithm for later antennas.

(iii) Burden of hardware elements: Multiple antennas at the same receiver need different length cables to temporally separate the signals received at each antenna in the CIR. Every 1 *nanosecond* separation in the CIR corresponds to about 1 *feet* of additional cable length (the exact number changes based on the cables dielectric
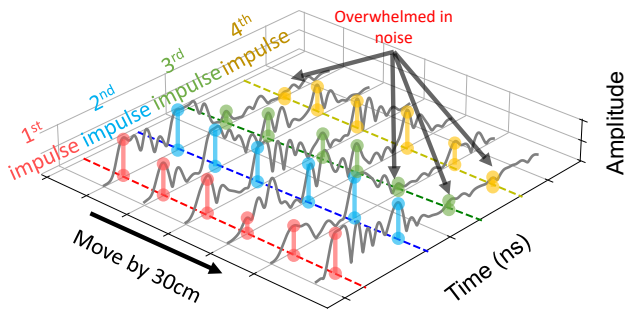
Figure 3: A four-antenna RX experiences significant dynamic range issue in CIR. Cannot be used for 3D tracking.
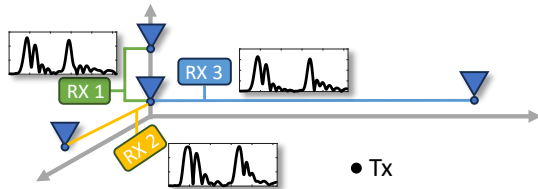

Figure 4: The 3-receiver hardware setup of UTrack3D.

properties). Additionally, because of high cable attenuation, one needs to try different combinations of amplifiers and attenuators to optimally tune the power of the obtained signals. Consequently, the complexity of the hardware significantly increases with the number of antenna elements. We therefore conclude that simply adding antenna elements to prior work in 2D tracking is not practical. We must modify the hardware design.

UTrack3D aims at creating a hardware setup resilient to dynamic range and multipath issues, while preserving simplicity and practicality. UTrack3D employs multiple 2-antenna receivers, rather than a single multi-antenna receiver. Doing so reduces the dynamic range issue to that between only two antennas for every RF-chain. Of course, this increases the hardware cost almost 3 fold, but given the inexpensive nature of the underlying COTS UWB devices, our costs are still at least an order of magnitude lower than optical tracking systems. Fig. 4 describes our tracking system. The infrastructure consists of three UWB receivers, each equipped with two antennas for phase differentiation. The antennas on the same receiver are connected using long RF cables. To reduce the hardware used, the first antenna is shared by all the receivers and is placed at the origin of a 3D area. When the TX sends a UWB message, this message is simultaneously received by all the UWB receivers. Each UWB receiver extracts CIR separately and streams to the host for computation. The host thus receives three CIRs corresponding to the signals arriving at each receiver, (see Fig. 4), which will be fed into the tracking pipeline for TX localization.

4.2 System Overview

We now present an overview of the various modules in UTrack3D, which we will describe in detail in Section 4.3.

Fig. 5 describes UTrack3D's processing pipeline at a high-level, which starts with obtaining the three CIRs. All CIRs originally have taps 1 *nanosecond* apart, but we upsample the complex CIR 64 times, as done by the hardware's leading edge detection algorithm [1]. The upsampled CIRs are used by the forward localization stage, which extracts the appropriate peaks from CIRs and perform precise tracking of the UWB transmitter using multi-lateration. However,

the results could be wrong when low-quality CIRs are captured, e.g., with encumbered with heavy multipath, or with extremely low SNR. The errors remaining after the forward localization will be detected and corrected in the posterior location check phase with an outlier detector and an optional re-initialization module.

At a high level, for each receiver, the forward localization stage proposes a few peak candidates in the CIR that are likely to be the line-of-sight (LOS) signals received by each antenna. Of course, only one of these pairs is valid. Therefore, the individual peaks are fed into a peak-pair ranking module to pair these candidates and sort them according to a likelihood criterion. The localization solver will extract phases from the top-ranked pair, calculate the phase difference change compared to the previous CIR and perform multi-lateration to determine the transmitter's location in 3D. This is one pass of the forward localization process. This location is then fed into an outlier detection module to determine if it has significantly deviated from the trajectory seen thus far. If an outlier is detected, the pipeline will loop back to the localization module, to select the next ranked peak pair for localization. This process will repeat until the estimated location is in the expected region. If no such location is found after traversing all peak pairs, UTrack3D will use part of the previous trajectory to predict a location as the current estimation. Thus, UTrack3D uses observed CIRs to compute the change in phase difference to track the location of the transmitter. However, merely tracking the phase changes cannot provide a reliable initial location. We explore if it is possible to obtain an approximate absolute location that can also be used for reinitialization if tracking is lost for too long and we lose count of phase wraping, also clearing any accumulated errors in the previous tracking process.

4.3 System Details

We will now discuss the specific challenges faced by each module and how UTrack3D tackles them.

4.3.1 Peak candidate proposals: The ideal CIR of a double-antenna receiver is a sequence of complex samples with only two prominent peaks, corresponding to the received signal at each antenna. Nevertheless, because of multipath, the CIR often has several peaks with comparable amplitude. Moreover, due to the non-uniform radiation pattern of the antenna, it is possible for a LOS peak to be lower than a multipath reflection. To prevent missing LOS peaks, UTrack3D proposes several peaks that could correspond to the LOS signal. However, proposing candidate peaks is non-trivial. Fig. 6 shows a sample CIR. Apart from the clearly-separated peaks (marked in green dash line), some peaks are "merged" with adjacent peaks (marked as red dash line). This happens when the propagation delay of two paths is very close. Missing a LOS peak could affect current and future localization (we will illustrate such dependency later). The question is: *How can we extract all the peaks, including the merged ones, that are likely to be LOS signals?*

The key observation is that even if two peaks are merged, it is still possible to extract each individual peak from the merged signal due to discontinuities introduced in the pulse shape (observe the pulse shape at the red dot in Fig. 6). If such discontinuity can be mathematically described, one will be able to extract the invisible peaks which are merged with other multipath signals. To achieve this, UTrack3D operates on the upsampled CIR signal and calculates
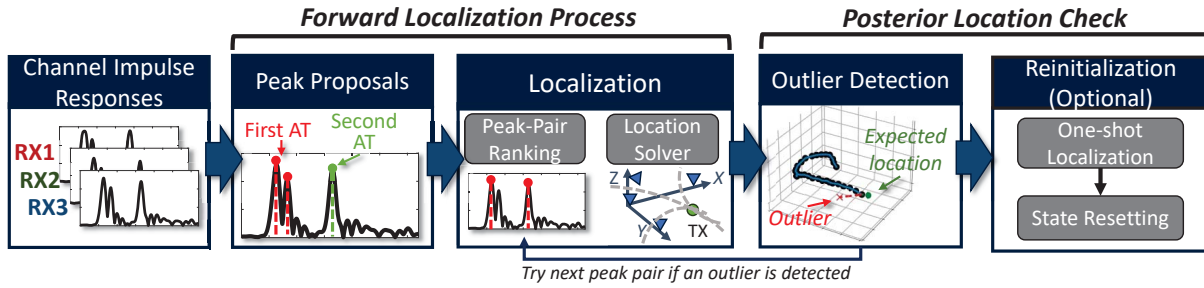
Figure 5: UTrack3D's system pipeline. Forward Localization Process is sufficient in most cases. Posterior checks prevent outliers.
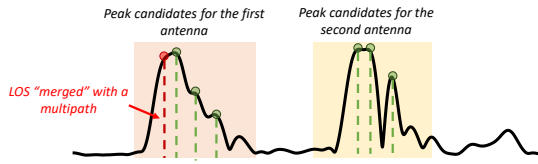

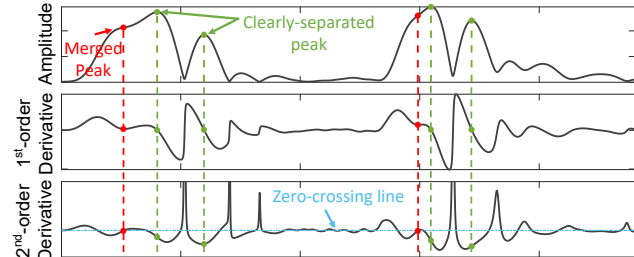Figure 6: A sample CIR with multipath. Peaks can be "merged" with an adjacent multipath.

Figure 7: A sample CIR with multipath and its first and second order derivative for detecting hidden peaks.
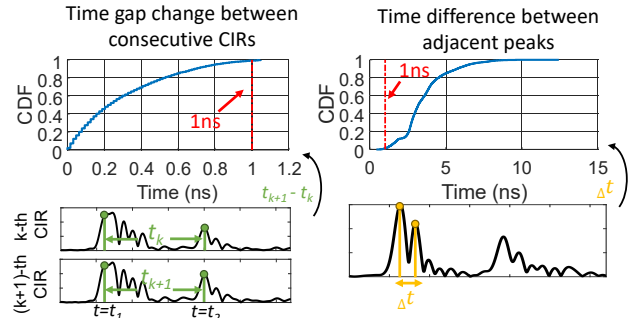

Figure 8: The time gap change of LOS signals between two consecutive packets is much smaller than the time difference between adjacent peaks within a CIR.

the 1st-order and 2nd-order derivative of the CIR magnitude. The 1st-order derivative describes the slope of the pulse's shape of the CIR. Intuitively, the CIR portion showing a small slope (flat in pulse shape) has a high probability to include a merged peak. On the other hand, the 2nd-order derivative gives the convexity of the CIR amplitude. We observe that the discontinuity of the pulse shape frequently occurs when the convexity of the CIR flips (i.e., the 2nd-order derivative crosses zero). Fig. 7 demonstrates the correlation between these two properties and a merged peak (see the red dots). UTrack3D identifies as a merged peak, the CIR sample which simultaneously satisfy: (i) the slope is smaller than a threshold, (ii) the convexity flips, and (iii) the amplitude of the original CIR is larger than a threshold.

The merged peaks, together with other clearly-separated peaks, form the candidate peaks. To reduce the processing overhead, UTrack3D preserves only the first n prominent peaks for each antenna, which are then fed into the localization module for peak-pair ranking and location calculation.

4.3.2 Localization: The localization module includes a peak-pair ranking algorithm, which sorts all the peak-pairs by the confidence of being LOS signals, and a location solver to estimate the transmitter's 3D coordinates.

Peak-pair ranking. UTrack3D pairs two peaks, one from the candidates for the first-antenna LOS signal and another from the

candidates for the second-antenna LOS signal, in this module. As mentioned, there are n peak candidates for each antenna, yielding n^2 combinations of peak pairs. Exhaustively trying every pair would break hopes of real-time tracking. To reduce the computation overhead, UTrack3D ranks all the peak-pairs according to the confidence of being LOS signals before feeding them into the location solver. We ask: *how to quantify the confidence of a peak pair?*

The first intuition is that since a LOS signal travels the least distance, the peaks with a lower tap index in the CIR are assigned a higher confidence. However, relying solely on arriving time proves inadequate. The multipath of the first antenna may appear just before the second peak candidates, causing peak-pair ranking to mistakenly assign a high confidence to this multipath peak.

UTrack3D leverages the fact that the time difference of two LOS signals in the CIR is nearly equal in two consecutive UWB packets. Mathematically, the time gap between two LOS peaks equals to the cable delay plus the in-air travelling time difference of two signals:

$$t_{gap} = t_{cable} + \Delta t_{air}, \qquad (5)$$

where the cable delay can be viewed as a constant. In two consecutive packets, the physical motion is very small, yielding a very minute change in Δt_{air}. Therefore, the change of their time gap between LOS signals is negligible. Fig. 8 shows the CDF of the time gap change between two consecutive UWB packets. Such time gap change is typically smaller than $1ns$. In contrast, the time difference of two adjacent peaks are much larger than $1ns$. If we mistakenly choose the wrong candidate, the inter-antenna time gap will have a sharp change: in Fig. 8, once the previous time gap t_k has already been determined, the best match for peak with $t = t_1$ will be the peak with $t = t_2$ for the $(k + 1)$-th packet. UTrack3D employs a

generalized normal distribution GN [44]:

$$w_{gap}(t_{k+1} - t_k, \beta) = \frac{\beta}{2\Gamma(1/\beta)} e^{\left(-\frac{|t_{k+1}-t_k|^\beta}{\sigma}\right)} \quad (6)$$

to model the effect of time gap change. Compared to the original normal distribution, GN distribution allows for tuning the weight of the main lobe on probability distribution function (PDF) using a parameter β. This property satisfies the demand to rank peak pairs: the peaks with near-zero time gap change will have quasi-equal confidence, while larger time gap change decreases the confidence. In our implementation, we empirically set $\sigma = 1ns$ and $\beta = 5$.

The final peak-pair ranking algorithm runs as follows: UTrack3D first calculates the confidence w_{gap} of each peak pair using the time gap change. Then all the peak pairs will be sorted by the ascending order of their response time. A decreasing confidence w_t will be assigned to the sorted peak pairs. The final confidence is calculated as: $w = w_t w_{gap}$. The peak pairs will be ranked using attribute w and are fed into the location solver for localization.

Location solver. Given a pair of peaks in the CIR, the location solver extracts the phase for each peak and subtracts them to get the inter-peak phase difference $_\Delta\phi^c$. The change of $_\Delta\phi^c$ is only caused by the air path difference from the transmitter to each antenna. Such change is minute between two consecutive UWB packets so UTrack3D wraps the phase change into $[-\pi, \pi)$. The phase difference change is mapped to the change of the air path difference using Equation (4). This builds a hyperbolic constrain to track the location of the transmitter. Given three hyperbolic constrains, UTrack3D is able to obtain a unique solution to the transmitter's location using L-BFGS-B optimization [11, 69]. UTrack3D sets the seed at the most recent location with $20cm$ bound around this searching point.

Of course, as the phase could wrap between two UWB packets, the maximum allowed moving speed in UTrack3D is bounded. Theoretically, the moving speed must be less than $\frac{\lambda R}{4}$ (R is the UWB packet rate) to ensure no phase wrapping. Since UTrack3D uses $4GHz$ UWB signals with a update rate of $77Hz$, the maximum moving speed is $1.45m/s$. Detailed evaluation is shown in Section 6, and we leave optimizations to improve update rate for future work.

4.3.3 Outlier Detection: The calculated location in the localization module, however, can be erroneous. This error could either result from choosing the incorrect peak pair, or receiving an occluded CIR. Without handling these cases correctly, such error will propagate to the following tracking because of UTrack3D's dependency on the previous tracking information.

UTrack3D employs an outlier detection algorithm to determine whether a location is an outlier, given the previous trajectory. The outlier detection is making prior prediction of the transmitter's current 3D coordinate using only the location and the velocity of the previous samples: $\hat{p}_{i+1} = p_i + v_i t$. The true location is expected to be located in a sphere centered at $\hat{p}_{i+1}$ with a radius representing the tolerance of prediction error:

$$|p_{i+1} - \hat{p}_{i+1}| \le \epsilon_{i+1}. \quad (7)$$

where error term ϵ_{i+1} is modeled as the sum of the moving average of previous prediction errors $\sum_{j=i-k}^{i-1} \theta_j \epsilon_j$ and a fixed Gaussian error distribution $\mathcal{N}(0, \sigma^2)$.

If an outlier is detected, the algorithm selects the next peak pair for location calculation. This process repeats until a valid location is found, or if no locations are valid, the prediction $\hat{p}^{i+1}$ is used.

4.3.4 (Re-)Initialization: The last unresolved question is *how is the initial location obtained?* Equation (4) illustrates why the initial location is needed: It provides a reference air path difference to compute per-CIR air path difference that follows. One trivial solution is to always trigger tracking from a known starting point, and periodically guide the user to move the object back to this initial location. However, we would like to avoid such requirements.

UTrack3D explores two options to automatically initialize the air path difference: Localization using time-difference-of-arrival (TDoA) or two-way ranging (TWR).

(1) TDoA-based localization: Apart from the phase difference, CIR also provides inter-antenna *time difference* t_{gap} which is influenced by the transmitter's location. According to Equation (5), the in-air propagation time difference of two antennas t_{air} can be directly obtained by subtracting the cable delay t_{cable} (calibrated beforehand) from t_{gap}. Then the air path difference is the product of the speed of light and in-air time difference, which can be directly used as a constrain to locate the transmitter.

(2) TWR-based localization: UTrack3D can employ two-way ranging protocol, as specified in [10], to locate the transmitter via multi-lateration by exchanging UWB messages with each receiver to measure the time of flight (TOF). The TX-RX distance is the product of ToF and the speed of light. Done right, TWR-based localization will not need to calibrate the cable delay. However, it complicates the communication protocol as it requires the infrastructure receivers to send message as well.

These two methods do not rely on historical positions, hence they can be used as a one-shot estimation of the initial location. Of course, because of the limited precision in determining peak locations, both TDoA-based and TWR-based methods yield high localization error, usually at decimeter-level. However, note that UTrack3D does not continuously use the time-difference-based estimation for localization—it is used only to estimate the initial location. The later localization is still performed using the phase-based method. Further, as a tracking application, UTrack3D does not care about the absolute localization errors, but concentrates on minimizing the bias-free trajectory errors. Detailed analysis of the effect of initial location error is in Section 6.

With one-shot localization, the user could trigger a reset from the transmitter itself, say by pushing a button indicating to the receivers to reset location.

5 IMPLEMENTATION

We prototype UTrack3D on COTS UWB devices. Fig. 9 shows the UTrack3D's prototype setup, including a simple UWB transmitter and one of the UWB receivers.

UWB transmitter (tracked in 3D). The UTrack3D transmitter which will be mounted on the object being tracked is quite simple with only a UWB chip and a microcontroller. It uses an Adafruit Feather M0 base design with an ATSAMD21G18 ARM Cortex M0 processor and a DWM1000 UWB chip [3]. The transmitter is only $5.7cm \times 2.1cm$, miniaturized to fit onto a wide range of accessories, like a stylus, a pen, or a carving knife. The UWB channel is set to
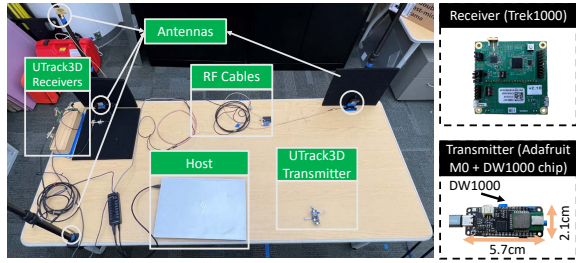
Figure 9: Table-top UTrack3D implementation.

UWB Channel 2, with $3.993GHz$ center frequency and a $500MHz$ bandwidth. At this channel, the UWB wavelength is $7.51cm$. To ensure a very high packet reception rate, we set the UWB data rate to $110Kbps$ (the lowest available data rate) and the preamble length to 2048 (one of the most reliable, in our testing). The UWB transmitter continuously sends UWB packets with only 2 bytes indicating the packet sequence. The UWB transmitting rate is about $77Hz$ and this data is processed real-time at the receivers and then at the host computer.

UWB receiving system (static infrastructure). The receiving system of UTrack3D consists of multiple double-antenna Trek1000 UWB receivers. To locate an object in 3D, at least three receivers are needed. In our minimal setup in Fig. 9, four antennas are connected to three Trek1000 boards. One antenna, at the left corner in the figure, is shared by all the boards, and each receiver independently connects to a second antenna along each of the three axes. Two antennas at the same receiver are connected using a long RF cable. As the signal arriving at the second antenna will experience attenuation when propagating in the RF cable, leading to disproportional received power between the first and second LOS signal, an RF attenuator is attached at the first antenna to balance the peak power. All the receivers stream data to a host with Intel Core i7-9750H and 16GB memory via serial port. The host creates an independent thread for data streaming from each receiver and synchronizes them in the main thread. The overall data streaming rate in our implementation is $77Hz$. A higher streaming rate can be available with fine-grained data compression and system-level optimizations. We leave it to future work.

6 EVALUATION

We evaluate UTrack3D in an indoor lab environment as shown in Fig. 10. Our setup has been built on top of a typical movable desk. Eight OptiTrack [6] IR cameras are placed around the tracking region to capture the ground-truth moving trajectory of the transmitter at $120Hz$, with a precision of $0.3mm$. For most of the experiments, the UWB transmitter moves within the rectangular cuboid formed by the four antennas over the desk.

6.1 UTrack3D Tracking Performance

We first evaluate the overall performance of UTrack3D in terms of 3D tracking accuracy. As illustrated in Section 4.3, the localization of UTrack3D involves two stages: (i) UTrack3D first calculates the location of the transmitter with a pair of candidate peaks in the *forward localization*, then (ii) performs *outlier detection* to re-select the peak pair if the estimated location significantly deviated.

We draw arbitrary 3D shapes by hand in the air. Movements are confined to a small volume over a standard lab desk (see Fig. 10). The transmitter is powered using a USB powerbank and we move

the transmitter by holding it at the USB-C connector, as shown in Fig. 11. We overlay the trajectory calculated by UTrack3D over the ground truth (after origin shift and axis rotation). The three IR markers at the transmitter, visible in Fig. 11 are used by the ground truth system. We test upwards of 20 trajectories to evaluate UTrack3D's performance.

Fig. 12 gives the visualization when we write a word ("Good") and when we draw a 3D geometrical shape composed of squares in different planes. A top-view and two side-views of the tracked geometric shape are shown at the bottom demonstrating the tracking accuracy. UTrack3D faithfully replicates the trajectory of the UWB transmitter, demonstrating UTrack3D's capability.

Quantitatively, Fig. 13 shows the CDF of tracking error in these two examples. Approximately 90% of the samples have an error smaller than $9mm$. This level of precision is sufficient for a wide range of applications requiring centimeter-level accuracy, like VR painting or tracking robot manipulation. As UTrack3D is a two-stage pipeline, we further evaluate the pipeline when only forward localization is enabled (FL-Only) with the most likely peak-pair to study the utility of outlier detection and trajectory prediction in UTrack3D. The CDF shows poorer performance without the full UTrack3D pipeline. This lower accuracy is the effect of a few intermediate outliers that offset the subsequent tracking. With no posterior checks, outliers are kept and errors propagate.

6.2 Micro-Benchmarks

6.2.1 Drift over time: Ideally, for phase difference ϕ_t at any time t, the accumulated phase difference change $\Delta\phi_t$ from the initial phase difference ϕ_0 satisfies: $_\Delta\phi_t = 2n\pi + wrap(\phi_t - \phi_0)$, where n gives the number of integer wraps and $wrap()$ is a phase modulo function mapping the phase into $[-\pi, \pi)$. When ϕ_t varies with Gaussian distribution (phase stability), the change in phase difference also conforms with Gaussian distribution (the integer portion of phase change can be explicitly determined and ignored during the tracking). Hence, theoretically UTrack3D should not drift over time if the peaks in CIR are clearly extracted.

To study whether UTrack3D will drift in practice, we place the transmitter on a robotic arm (Elephant myCobot 280) and repeat a simple one-stroke trajectory for 10 minutes (about 50 cycles). To minimize the influence of locations on tracking accuracy, the robotic arm only sweeps around a small region ($30cm$ in extent). The drift error is calculated by measuring the deviation of all the samples of later cycles with respect to the first cycle. Fig. 14 shows that, even after 10 minutes, the mean, $90\%tile$, and the maximum error remain at $2mm$, $5mm$ and $8mm$ respectively. No apparent drift is observed during this process. Our evaluation demonstrates that long-term tracking has little effect on the tracking accuracy of UTrack3D. This superior property compared to inertial sensing opens up various opportunities, but a comparison study is out of scope since we focus on a UWB-only solution. For instance, a sculptor can digitize the motion of the sculpting tool for a long time without requiring frequent re-initialization, enabling a more natural activity for the sculptor.

6.2.2 Effect of the TX location: UTrack3D solves for the intersection of hyperbolas through an optimization method. This inevitably results in an error distribution dependent on the transmitter's location, since a small error in distance difference estimate can have
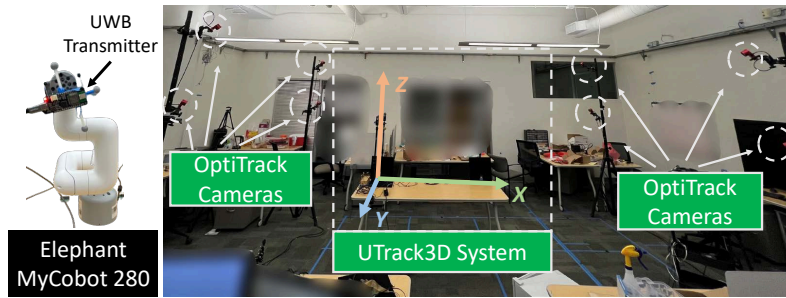
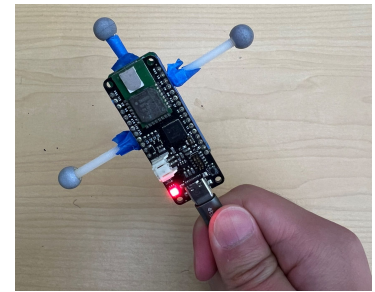Figure 10: The evaluation setup of UTrack3D. Ground-truth is acquired using Opti-Track system [6].

Figure 11: The transmitter is held by hand at the USB power connector.
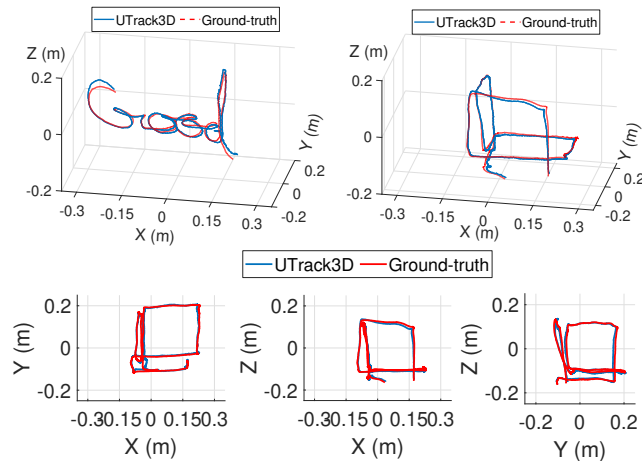

Figure 12: 3D trajectory estimated by UTrack3D and the ground truth: (top-left) Letters; (top-right) Geometric shape; (bottom) The cross section view of geometric shape.
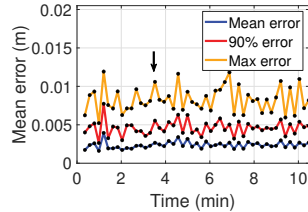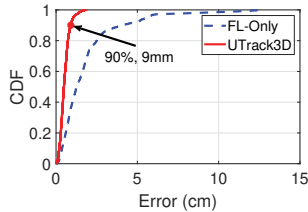

Figure 13: The tracking error CDF of UTrack3D.

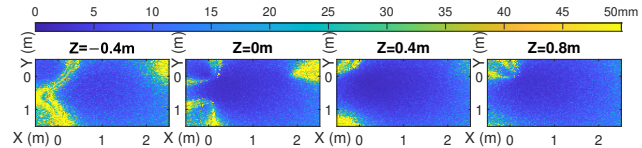Figure 14: UTrack3D shows no drift over time.

Figure 15: Simulated spatial error heatmap of UTrack3D using real-world parameters. First heatmap is below the table.

Figure 16: Simulation setup: Locations of receiving antennas and transmitters.

Table 1: Simulation parameters: Locations of received antennas.

	Antenna Positions (AT1~ AT4)
Co-planar	$(0, 0, 0)$, $(1.5, 0, 0)$, $(0, 0.8, 0)$, $(1.5, 0.8, 0)$
One-Z-Diff	$(0, 0, 0)$, $(1.5, 0, 0)$, $(0, 0.8, 0)$, $(1.5, 0.8, 0.8)$
All-Z-Diff-1	$(0, 0, 0)$, $(1.5, 0, 0.1)$, $(0, 0.8, 0.5)$, $(1.5, 0.8, 0.8)$
All-Z-Diff-2	$(0.02, -0.04, 0.02)$, $(1.5, -0.04, 0.28)$, $(0.07, 0.71, -0.25)$, $(1.41, 0.98, 0.75)$

mathematical problem, we simulate the GDoP in Python by feeding the parameters of our setup and expected phase errors. The phase difference change is modeled as a Gaussian distribution, with a standard deviation measured with the real-world data $(4°)$. In the simulation, we repeat the same trajectories with equal Gaussian variance in phase measurement at different locations. Fig. 15 shows the heatmap of median error when we scan the whole XY-plane at a height of $-0.4m$ (below the table), $0m$, $0.4m$ and $0.8m$. Errors larger than $50mm$ are clipped to $50mm$.

We make several important observations. First, UTrack3D achieves the best performance when the transmitter is located in the central region surrounded by the antennas, where the error is typically $< 1cm$. Moving the transmitter further from the anchors will in general degrade the tracking accuracy. Second, the region very close to each antenna produces significantly higher errors due to poor GDoP. Even slight variance incurs significant errors $(> 5cm)$ in localization. One solution is to add an additional receiver with antennas at a different location; when the transmitter is near an antenna, the remaining receivers can compensate the error, and still robustly track the transmitter's location. Evaluation with additional antennas is discussed next.

substantially different localization error depending on whether that error occurs near the asymptote or near the center of the hyperbola. This effect is known as the geometric dilution of precision (GDoP).

In the minimum setup used in UTrack3D, four antennas are placed at four corners of the tracking region (origin and the extremes of the three axes). Taking the coordinate system in Fig. 10, the positions of antennas are $(0, 0, 0)$, $(1.5, 0, 0)$, $(0, 0.8, 0)$, $(0, 0, 0.8)$, respectively. The GDoP is tightly dependent on the relative location of the transmitter with respect to these antennas. Since it is a pure
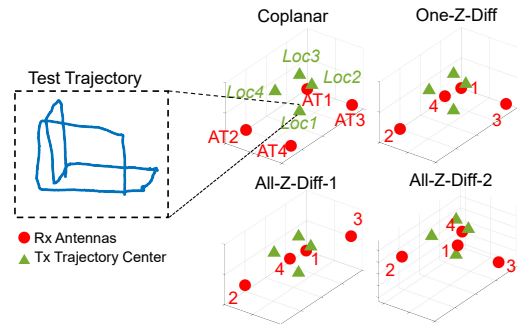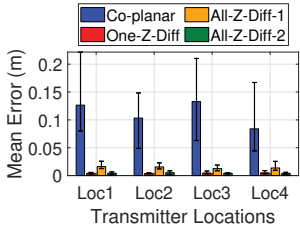
Figure 17: The tracking error when receiving antennas are placed differently.
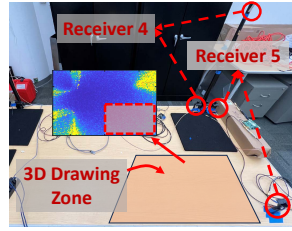
Figure 18: New antenna positions and the drawing zone.

Figure 19: The tail error is improved when using more receivers.

Figure 20: The effect of movement speed and phase wraps (PW) observed.
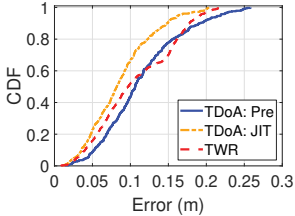

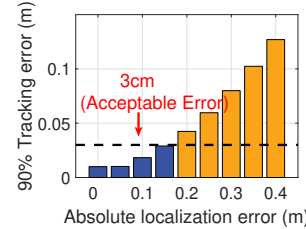**Figure 21: Absolute localization error of performing re-initialization.**

Figure 22: The effect of absolute localization error on tracking accuracy.

6.2.3 Effect of the locations of receiver antennas: Apart from the transmitter's trajectory and location, the locations of the fixed receiver antennas may also influence the geometric dilution of precision, affecting the tracking accuracy. Understanding this effect can guide the deployment of receiver antennas to optimize UTrack3D's performance for certain expected trajectories. Particularly, we are interested in answering the question: *Will diversifying coordinates of receiver antennas improve observed precision of 3D tracking?*

We evaluate the effect of receiver antennas' locations on the accuracy of tracking using a Python simulation based on underlying phase precision values from practical real-world trajectories. In UTrack3D's minimum setup, there is exactly one antenna which has a different X, Y or Z coordinate from the other three. For example, the antenna placed at $(0, 0, 0.8)$ is the only one which is not located in the $Z = 0$ plane. Relaxing this constraint in simulation, we compare the performance of four different setups:

(1) Nearly co-planar: All the four antennas are approximately located in the same plane (Fig. 16 (top-left));

(2) One-Z-Diff: Three antennas are placed at plane $Z = 0$, one antenna is above the other three (Fig. 16 (top-right)). This is also the setup UTrack3D uses for other evaluations;

(3) All-Z-Diff-1: None of the four antennas have the same Z coordinates (Fig. 16 (bottom-left)). This is done by moving antenna 2 and 3 vertically based on One-Z-Diff setup. Note that diversifying Z in this way will shrink the convex hull formed by antennas.

(4) All-Z-Diff-2: None of the four antennas have the same Z coordinates (Fig. 16 (bottom-right)). This is done by rotating One-Z-Diff setup by a certain angle. Diversifying Z coordinate in this way does not change the size of the convex hull formed by antennas.

The exact antenna locations are provided in Table 1. We use the real-word trajectory (Fig. 12 (top-right)) for our evaluation. Because the transmitter's location also influences the tracking error (as seen in Section 6.2.2), we repeat the test four times by varying the centroid of the trajectory (centered at the various green triangles shown in Fig. 16). The centroids of $Loc1$ to $Loc4$
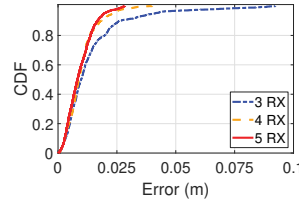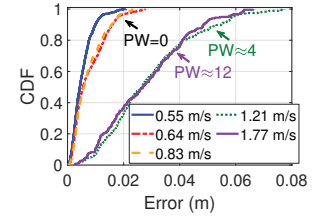
are $(0.6, 0.3, 0)$, $(0.3, 0.3, 0.3)$, $(0.6, 0.3, 0.6)$ and $(1.2, 0.3, 0.6)$, respectively. The other parameters of these trajectories are exactly the same. Fig. 17 presents the 90%*tile* tracking error of each receiver setup. Co-planar setup gives the worst performance. This is because zero-diversity in Z coordinate results in poor geometric dilution of precision. Interestingly, we also observe clear performance degradation in All-Z-Diff-1 setup, even if Z coordinates are further diversified. Such degradation comes from the fact that the convex hull formed by All-Z-Diff-1 is smaller, compared to One-Z-Diff setup. The effect of the size of the convex hull is more clear when we observe the fourth setup, All-Z-Diff-2. Similar to All-Z-Diff-1, the Z coordinates of its antennas are all diversified. However, since All-Z-Diff-2 is simply a rotation of One-Z-Diff, the size of the convex hull does not change. As a result, All-Z-Diff-2 clearly outperforms All-Z-Diff-1. Finally, we observe that All-Z-Diff-2 has similar error to One-Z-Diff, which demonstrates that merely diversifying Z coordinates does not further improve the accuracy.

6.2.4 Effect of the number of receivers: While a minimal setup needs only 3 dual-antenna receivers, having more receivers expands the coverage of UTrack3D's tracking region, and can also improve performance when the transmitter is already within a good tracking region. To add two more receivers to our original setup. The additional receivers are placed as shown in Fig. 18. We choose a location within the tracking region where the benefit of adding additional antennas is clear. At this location, the 3-antenna minimal setup has around 2 *cm* 90%*tile* localization error. All the 5 receivers simultaneously receive UWB signals and stream CIR to the host. However, only a subset of the receivers will be used to run UTrack3D's tracking algorithm. Fig. 19 shows the CDF of errors when 3 (removing RX4 and RX5), 4 (removing RX 5), and 5 receivers are used in tracking. We observe improved tail-accuracy. At the 90%*tile*, we observe a slight improvement (2*cm*, 1.9*cm* and 1.7*cm* with 3, 4 and 5 receivers). However, the worst-case errors are significantly reduced from 9.2*cm* with 3 receivers to 2.8*cm* with 5 receivers. In summary, more receivers provide better tracking accuracy though increase the hardware and processing costs.

6.2.5 Effect of moving speeds: Since UTrack3D employs phase difference change to track motion, two consecutive CIRs should have a phase difference change less than π. Otherwise, a phase wrapping error can occur which will degrade the tracking performance. UTrack3D's theoretical maximum moving speed at which the performance is not influenced by phase wrapping is computed as $\frac{\lambda R}{4}$, where λ and R are the UWB wavelength and the UWB packet rate. In UTrack3D the UWB packet rate is 77Hz, which enables a 1.45m/s moving speed if using 3.993GHz UWB channel. However, the practical maximum speed is typically lower because of phase

Table 2: Time consumption of each module in UTrack3D.

Procedure	Sub-Module	Time (ms)
Send/Receive UWB	N/A	$4.0 \sim 13.0$ ($77 \sim 250Hz$)
Process Data (i7-9750H CPU)	CIR Reading	2
	Peak Proposing	4.3
	Localization	2.9
	Outlier Detection	<1
	(Re-)initialization	2.2

Table 3: The effect of blocking on UTrack3D.

	Acrylic Board	Wood	Human Body	Metal Sheet
Signal Attenuation	0.9dB	2.4dB	11.5dB	14.8dB
$90\%tile$ Error	$1.6cm$	$1.3cm$	×	×
CIR Available?	Yes	Yes	Yes	No

variance. We ask: *How does the moving speed influence the tracking performance.* To evaluate this, we repeat the same trajectory with different moving speeds. Fig. 20 presents the error CDF when the transmitter is moving at different speeds ("PW" is the estimated phase wrapping count.). For a moving speed of 0.32 m/s, 0.44 m/s and 0.65m/s, UTrack3D achieves near-optimal performance with $90\%tile$ error < 1cm. To give an intuition, these three speeds correspond to slow writing, normal writing, and fast writing in the air. When the moving speed rises up to 1.14m/s, an error in even one UWB transmission starts to increase the chance that phase wrapping occurs and thereby increases the overall error. Moving at such a speed typically involves some sudden and intense motion, e.g., a rapid stroke of the hand, and is unlikely. Note that UTrack3D is after all a table-top solution; a speed of 0.65 m/s would approximately move the object the entire width of the table in slightly over 1 second, and the theoretical max speed of 1.14m/s would move the object across the entire length of the table within that time. For example, digitizing a carving process only needs a velocity less than 0.2m/s, which is perfectly covered by UTrack3D. Additionally, while we test with a packet rate of 77Hz, a packet rate as high as 250Hz is available by increasing the UWB data rate (to 6.8$Mbps$). Therefore, the maximum speed with no phase wrapping can reach up to 4.7m/s, satisfying the demands of most applications.

6.2.6 Effect of initialization: UTrack3D provides an option to automatically compute the initial position using time difference in the CIR. In this evaluation, we study *the performance of this auto-initialization algorithm* and *how the initialization error will influence the tracking performance.*

Initial localization error. The initial location is estimated using TDoA or TWR method. For TDoA method, we measure the on-cable delay using two approaches: (i) Computing cable delay in advance (pre-computing (PRE)); and (ii) Computing cable delay using samples in a previous time window, typically a few seconds ago (just-in-time computing (JIT)). Compared to pre-computing which strictly needs no prior information, just-in-time computing still needs calibration at the very beginning, but can dynamically adapt to the influence of environmental factors such as temperature affecting cable delay. For TWR method, an RF switch is used to disable the signal arriving at the first antenna. Fig. 21 gives the error CDF of the absolute localization error. The median absolute

localization error of PRE- and JIT-computing for cable delay calibration is 8.2cm and 10.6cm respectively, consistent with typical UWB localization values. Overall, JIT computing outperforms PRE computing because it captures the temporal environmental factors which can influence the cable delay. The median error for TWR-based approach is about 9.6cm providing a baseline to compare against. The advantage of TWR-based method is that the localization is not affected by the cable delay, at a cost of more complicated communication protocol. Note that *the evaluated absolute localization error is not the final error for UTrack3D*; it is only the deviation of the first sample from the ground-truth. As a tracking system, UTrack3D is rather insensitive to initial bias, though a non-linear mapping between phase-differences and absolute location will affect UTrack3D's tracking performance. We explore this non-linear relationship next.

Initial localization error's effect on tracking performance. According to Eq. 4, an initial localization error should have no influence on phase difference change measurement, but will corrupt the location estimation by polluting the last sample's in-air distance difference. To illustrate, imagine two transmitters located at different positions p_1 and p_2. Even if they observe the same phase difference change after moving to p_1' and p_2' respectively, the relative distance $p_1' - p_1$ does not equal $p_2' - p_2$. To understand the effect of this non-linear mapping on tracking, we virtually deviate the coordinate of the first location. Then, the estimated trajectory is re-aligned to the ground-truth to decouple the effect of the initial bias and just compare the tracking error.

Fig. 22 shows the relation between the $90\%tile$ tracking error and the initial localization error. The tracking error increases marginally from 9mm to 19.8cm as the absolute localization error increases from 0 to 50cm. Using our real-world measurements of the initial localization error in Fig. 21, we expect 1.2cm $90\%tile$ tracking error if performing initial localization using JIT measurements, or 2.9cm $90\%tile$ tracking error if using per-computed measurements. Averaging a set of initial locations can further improve the performance.

6.2.7 Effect of Occlusions: One advantage of UTrack3D over camera-based tracking solutions is that tracking can still be performed when the line-of-sight is blocked. We use different blocking materials (acrylic board, wood, human body, and metal) to test how signal blocking influences UTrack3D's performance. Table 3 presents the signal attenuation and the $90\%tile$ error. Acrylic boards and wood boards, two common packaging materials, have little influence on UTrack3D's performance. The LOS peak of each antenna is still clearly distinguishable. However, objects with a significant attenuation factor can lead to tracking failure. When a human body blocks, the signal peaks are still observable but can be overwhelmed by noise. Frequent such errors cannot be corrected. Finally, when blocking with a metal sheet, the receiver can lose CIR completely which shows a failure case and UTrack3D cannot function under such blockage.

6.3 System Evaluation

6.3.1 Processing Time: UTrack3D aims to track the motion of an object in real-time. The end-to-end tracking involves two parallel steps: (1) Sending UWB signals from TX and receiving at RX; (2) Processing the received UWB signals to estimate the transmitter's location. Table 2 shows the time consumption of each procedure.

The time delay of sending and receiving UWB signals is typically between 4 *ms* and 13 *ms*, governed by the UWB bitrate. The mandatory data processing procedures include the time spent on CIR reading (2*ms*), peak proposals (4.3*ms*), localization (2.9*ms*) and outlier detection ($< 1ms$). Note that while the localization and outlier detection can repeat several times until finding a location in the predicted region, in our experience, less than 5% samples need more than one loop to locate the correct peaks under UTrack3D peak-pair ranking algorithm, which significantly improves the efficiency of data processing. Re-initialization takes about 2.2*ms*. It only happens infrequently when auto-initialization is enabled to obtain the first location. Overall, UTrack3D takes about 10*ms* to estimate a location end-to-end after parallelizing UWB signals sending/receiving and data processing. This time delay currently enables a data rate of 77*Hz* constrained by the UWB transmission time, which is sufficient for a wide range of real-time applications.

6.3.2 Power consumption: The power consumption influences how often the tracked object will need to be recharged. When the UWB packet rate is 77*Hz*, the power consumption at the transmitter and one receiver is 210*mW* and 710*mW* respectively. Admittedly, hundreds of milliwatts might seem high for an embedded system. This is partly because UTrack3D receivers are built on a full evaluation kit (TREK1000) integrated with additional functions such as a screen, and the transmitter uses older version UWB module (DWM1000) with limited power optimizations. According to the datasheet [2], the power consumption can be reduced to 64*mW* for transmitting and 170*mW* for receiving by replacing DWM1000 with DWM3000 module, though we have not practically tested it. We leave this optimization for future work.

6.3.3 Cost: Our infrastructure setup comprises 3 TREK1000 boards which are the most costly components in UTrack3D. Note that TREK1000 is an evaluation kit which we use in UTrack3D since it has an SMA port that provides flexibility of connecting our own antennas, RF cables, and attenuators. Together these cost around $600. In contrast, our transmitter, which is based on our custom design costs around $50, which is equivalent in capability to the TREK1000 boards, except for the SMA connector. Thus, we expect the cost of the UWB devices (3RX+1TX) to be around $200 when manufactured anew. External attenuators and cables add around $100 creating a $500 setup. Our software processing pipeline will be made open source and therefore will be available for free. Overall, UTrack3D significantly reduces the cost barrier to entry for 3D motion capture.

7 DISCUSSION AND FUTURE WORK

UTrack3D addresses the fundamental problems of tracking movements in 3D in a small confined space. However there will be a significant amount of effort needed if one wishes to convert UTrack3D into a commercial product. We discuss some of the current limitations and provide pointers to facilitate future work in this space.

The influence of dynamic ranging. The amplitude ratio of the two peaks at each antenna dynamically changes when the transmitter moves. In locations where one peak is overwhelmed by noise, UTrack3D loses tracking. This dynamic range problem is the fundamental factor limiting the UTrack3D's tracking range. One option is adding redundancy by using additional receivers. The

host could run an adaptive algorithm to analyze the confidence of the CIR provided by each receiver based on the peak amplitude and inter-peak time difference. Once a CIR's confidence falls below a threshold, the host would disregard this CIR. With the remaining CIRs, the host would still be able to track the object's movements.

Velocity limit. The maximum non-wrapping moving speed of UTrack3D is determined by both the sampling rate R and wavelength λ. With UTrack3D's setup, the upper-bound speed of non-wrapping is approximately 1.45*m/s*. This upper-bound allows tracking common 3D actions like carving, drawing, and hand gestures, but may fall short for actions needing high-speed motion. Remaining within the IEEE standard [10], one can push the upper bound of the non-wrapping velocity by increasing the sampling rate. Some potential improvements include using a shorter preamble for UWB or extracting fewer CIR samples. We leave these for future work.

Multi-object tracking. The current setup of UTrack3D only supports tracking one transmitter. To track multiple objects, all transmitters will need a round-robin protocol which emits UWB signals in sequence. This approach inevitably decreases the rate of sending/receiving UWB signals. However, one can also invert the trasmitter and receiver roles in UTrack3D, i.e., three fixed double-antenna transmitters send UWB signals to the UWB receiver attached on the object. A micro-controller will run UTrack3D tracking algorithm on the object. This setup will support tracking unlimited objects, so long as each object has enough compute-power to run the location solver. It is unknown whether the computing power of the tiny micro-controller on the object will be sufficient to perform the tracking task. The exploration of the possibility of multi-object tracking is left for future work.

Absolute localization. UTrack3D achieves millimeter-level accuracy in tracking 3D movements (relative localization of a sequence of 3D locations), but accurate absolute localization remains a challenge. The fusion of UWB with other sensors, such as inertial sensors, is a potential solution. Advanced fusion algorithms can preserve history to correct errors after a trajectory is complete, which improves the performance of absolute localization. We leave such post-processing for future work.

8 CONCLUDING REMARKS

UTrack3D provides a cost-effective table-top alternative for capturing 3D movements in a small area. We have developed a UWB-based tracking system that benefits from UWB's superior phase resolution and higher bandwidth, for sub-centimeter localization. UTrack3D makes architectural innovations in creating a multi-UWB dual-antenna system and algorithmic innovations via our signal processing pipeline to provide a robust 3D tracking solution which can benefit a wide variety of applications. Indeed, we have only developed lab prototypes, and some effort will be required to transition it into commercial products, however, we believe we have laid solid foundations for future research and product development in this space.

REFERENCES

[1] DW1000 user manual. https://www.qorvo.com/products/d/da007967.
[2] DW3000. https://www.qorvo.com/products/p/DWM3000#documents.
[3] DWM1000. https://www.qorvo.com/products/p/DWM1000.
[4] Intel Realsense. https://www.intelrealsense.com/.
[5] Microsoft Kinect. https://www.microsoft.com/en-us/d/azure-kinect-dk/8pp5vxmd9nhq/.
[6] OptiTrack. https://optitrack.com/.
[7] Qualisys. https://www.qualisys.com/.
[8] Vicon. https://www.vicon.com/.
[9] ZED X. https://www.stereolabs.com/.
[10] IEEE standard for low-rate wireless networks. *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)*, pages 1–709, 2016.
[11] Richard H Byrd, Peihuang Lu, Jorge Nocedal, and Ciyou Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208, 1995.
[12] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.
[13] Gaoshuai Cao, Kuang Yuan, Jie Xiong, Panlong Yang, Yubo Yan, Hao Zhou, and Xiang-Yang Li. Earphonetrack: involving earphones into the ecosystem of acoustic motion tracking. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, pages 95–108, 2020.
[14] Yanjun Cao, Chenhao Yang, Rui Li, Alois Knoll, and Giovanni Beltrame. Accurate position tracking with a single uwb anchor. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 2344–2350. IEEE, 2020.
[15] Yifeng Cao, Ashutosh Dhekne, and Mostafa Ammar. Itracku: tracking a pen-like instrument via uwb-imu fusion. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, pages 453–466, 2021.
[16] Haige Chen and Ashutosh Dhekne. Pnploc: Uwb based plug & play indoor localization. In *2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8. IEEE, 2022.
[17] Ka C Cheok, Micho Radovnikovich, PK Vempaty, Gregory R Hudas, James L Overholt, and Paul Fleck. Uwb tracking of mobile robots. In *21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 2615–2620. IEEE, 2010.
[18] David Chiasson, Yuan Lin, Manon Kok, and Peter Shull. Asynchronous hyperbolic uwb source-localization and self-localization for indoor tracking and navigation. *IEEE Internet of Things Journal*, 2023.
[19] Ashutosh Dhekne, Ayon Chakraborty, Karthikeyan Sundaresan, and Sampath Rangarajan. TrackIO: Tracking first responders inside-out. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, pages 751–764, 2019.
[20] Mahanth Gowda, Ashutosh Dhekne, Sheng Shen, Romit Roy Choudhury, Lei Yang, Suresh Golwalkar, and Alexander Essanian. Bringing iot to sports analytics. In *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, pages 499–513, 2017.
[21] Mahanth Gowda, Ashutosh Dhekne, Sheng Shen, Romit Roy Choudhury, Sharon Xue Yang, Lei Yang, Suresh Golwalkar, and Alexander Essanian. Iot platform for sports analytics. *GetMobile: Mobile Computing and Communications*, 21(4):8–14, 2018.
[22] Mahanth Gowda, Justin Manweiler, Ashutosh Dhekne, Romit Roy Choudhury, and Justin D Weisz. Tracking drone orientation with multiple gps receivers. In *Proceedings of the 22nd annual international conference on mobile computing and networking*, pages 280–293, 2016.
[23] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.
[24] Bernhard Großwindhager, Michael Stocker, Michael Rath, Carlo Alberto Boano, and Kay Römer. Snaploc: An ultra-fast uwb-based indoor localization system for an unlimited number of tags. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, pages 61–72, 2019.
[25] Jose A. Gutierrez, Edgar H. Callaway, and Raymond L. Barrett. *IEEE 802.15.4 Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensor Networks*. IEEE, 2003.
[26] Zijun Han, Zhaoming Lu, Yawen Chen, Xiangming Wen, and Jiawei Jiang. Magicinput: Virtual handwriting interface using ubiquitous wifi signals. *IEEE Transactions on Industrial Informatics*, 2022.
[27] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
[28] Stefan Hinterstoisser, Vincent Lepetit, Slobodan Ilic, Stefan Holzer, Gary Bradski, Kurt Konolige, and Nassir Navab. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In *ACCV*, pages 548–562. Springer, 2013.

[29] Chengkun Jiang, Yuan He, Songzhen Yang, Junchen Guo, and Yunhao Liu. 3d-omnitrack: 3d tracking with cots rfid systems. In *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*, pages 25–36, 2019.
[30] Manikanta Kotaru and Sachin Katti. Position tracking for virtual reality using commodity wifi. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 68–78, 2017.
[31] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *ECCV*, pages 574–591. Springer, 2020.
[32] Chi Li, Jin Bai, and Gregory D Hager. A unified framework for multi-view multi-class object pose estimation. In *ECCV*, pages 254–269, 2018.
[33] Manolis Lourakis and Xenophon Zabulis. Model-based pose estimation for rigid objects. In *International conference on computer vision systems*, pages 83–92. Springer, 2013.
[34] Wenhan Luo, Junliang Xing, Anton Milan, Xiaoqin Zhang, Wei Liu, and Tae-Kyun Kim. Multiple object tracking: A literature review. *Artificial intelligence*, 293:103448, 2021.
[35] Zhihong Luo, Qiping Zhang, Yunfei Ma, Manish Singh, and Fadel Adib. 3d backscatter localization for {Fine-Grained} robotics. In *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, pages 765–782, 2019.
[36] Mohamed R Mahfouz, Michael J Kuhn, Yazhou Wang, Jonathan Turnmire, and Aly E Fathy. Towards sub-millimeter accuracy in uwb positioning for indoor medical environments. In *2011 IEEE Topical Conference on Biomedical Wireless Technologies, Networks, and Sensing Systems*, pages 83–86. IEEE, 2011.
[37] Mohamed R Mahfouz, Cemin Zhang, Brandon C Merkl, Michael J Kuhn, and Aly E Fathy. Investigation of high-accuracy indoor 3-d positioning using uwb technology. *IEEE Transactions on Microwave Theory and Techniques*, 56(6):1316–1330, 2008.
[38] Wenguang Mao, Jian He, and Lili Qiu. Cat: high-precision acoustic motion tracking. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 69–81, 2016.
[39] Matteo Menolotto, Dimitrios-Sokratis Komaris, Salvatore Tedesco, Brendan O'Flynn, and Michael Walsh. Motion capture technology in industrial applications: A systematic review. *Sensors*, 20(19):5687, 2020.
[40] Kevin Minne, Nicola Macoir, Jen Rossey, Quinten Van den Brande, Sam Lemey, Jeroen Hoebeke, and Eli De Poorter. Experimental evaluation of uwb indoor positioning for indoor track cycling. *Sensors*, 19(9):2041, 2019.
[41] Anastasios I Mourikis and Stergios I Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE international conference on robotics and automation*, pages 3565–3572. IEEE, 2007.
[42] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.
[43] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.
[44] Saralees Nadarajah. A generalized normal distribution. *Journal of Applied statistics*, 32(7):685–694, 2005.
[45] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. Fingerio: Using active sonar for fine-grained finger tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1515–1525, 2016.
[46] Adeshina Sirajdin Olagoke, Haidi Ibrahim, and Soo Siang Teoh. Literature survey on multi-camera system and its application. *IEEE Access*, 8:172892–172922, 2020.
[47] José Pino-Ortega and Markel Rico-González. Review of ultra-wide band in team sports. In *Innovations in Ultra-Wideband Technologies*. IntechOpen, 2020.
[48] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.
[49] Lanxin Qiu, Zhangqin Huang, Shaohua Zhang, Cheng Jing, Hao Li, and Shuyao Li. Multifrequency phase difference of arrival range measurement: Principle, implementation, and evaluation. *International Journal of Distributed Sensor Networks*, 11(11):715307, 2015.
[50] Sai Deepika Regani, Chenshu Wu, Beibei Wang, Min Wu, and KJ Ray Liu. mmwrite: passive handwriting tracking using a single millimeter-wave radio. *IEEE Internet of Things Journal*, 8(17):13291–13305, 2021.
[51] Nirupam Roy, Haitham Hassanieh, and Romit Roy Choudhury. Backdoor: Making microphones hear inaudible sounds. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 2–14, 2017.
[52] Stefan Schmid, Giorgio Corbellini, Stefan Mangold, and Thomas R Gross. Led-to-led visible light communication networks. In *Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing*, pages 1–10, 2013.
[53] Johannes L Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. Semantic visual localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6896–6906, 2018.

[54] Longfei Shangguan and Kyle Jamieson. Leveraging electromagnetic polarization in a two-antenna whiteboard in the air. In *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, pages 443–456, 2016.

[55] Anran Wang and Shyamnath Gollakota. Millisonic: Pushing the limits of acoustic motion tracking. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–11, 2019.

[56] Ju Wang, Liqiong Chang, Omid Abari, and Srinivasan Keshav. Are rfid sensing systems ready for the real world? In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 366–377, 2019.

[57] Jue Wang, Deepak Vasisht, and Dina Katabi. Rf-idraw: Virtual touch screen in the air using rf signals. *ACM SIGCOMM Computer Communication Review*, 44(4):235–246, 2014.

[58] Wei Wang, Alex X Liu, and Ke Sun. Device-free gesture tracking using acoustic signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 82–94, 2016.

[59] Teng Wei and Xinyu Zhang. mtrack: High-precision passive tracking using millimeter wave radios. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 117–129, 2015.

[60] Chenshu Wu, Feng Zhang, Yusen Fan, and KJ Ray Liu. Rf-based inertial measurement. In *Proceedings of the ACM Special Interest Group on Data Communication*, pages 117–129. 2019.

[61] Dan Wu, Ruiyang Gao, Youwei Zeng, Jinyi Liu, Leye Wang, Tao Gu, and Daqing Zhang. Fingerdraw: Sub-wavelength level finger motion tracking with wifi signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–27, 2020.

[62] Ning Xiao, Panlong Yang, Xiang-Yang Li, Yanyong Zhang, Yubo Yan, and Hao Zhou. Milliback: Real-time plug-n-play millimeter level tracking using wireless backscattering. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–23, 2019.

[63] Jing Yang, BaiShun Dong, and Jiliang Wang. Vuloc: Accurate uwb localization for countless targets without synchronization. *Proceedings of the ACM on Interactive,*

Mobile, Wearable and Ubiquitous Technologies, 6(3):1–25, 2022.

[64] Lei Yang, Yekui Chen, Xiang-Yang Li, Chaowei Xiao, Mo Li, and Yunhao Liu. Tagoram: Real-time tracking of mobile rfid tags to high precision using cots devices. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 237–248, 2014.

[65] Sangki Yun, Yi-Chao Chen, and Lili Qiu. Turning a mobile device into a mouse in the air. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 15–29, 2015.

[66] Sangki Yun, Yi-Chao Chen, Huihuang Zheng, Lili Qiu, and Wenguang Mao. Strata: Fine-grained acoustic-based device-free tracking. In *Proceedings of the 15th annual international conference on mobile systems, applications, and services*, pages 15–28, 2017.

[67] Cemin Zhang, Michael Kuhn, Brandon Merkl, Mohamed Mahfouz, and Aly E Fathy. Development of an uwb indoor 3d positioning radar with millimeter accuracy. In *2006 IEEE MTT-S International Microwave Symposium Digest*, pages 106–109. IEEE, 2006.

[68] Cheng Zhang, Qiuyue Xue, Anandghan Waghmare, Sumeet Jain, Yiming Pu, Sinan Hersek, Kent Lyons, Kenneth A Cunefare, Omer T Inan, and Gregory D Abowd. Soundtrak: Continuous 3d tracking of a finger using active acoustics. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):1–25, 2017.

[69] Ciyou Zhu, Richard H Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on mathematical software (TOMS)*, 23(4):550–560, 1997.

[70] Yuzhou Zhuang, Yuntao Wang, Yukang Yan, Xuhai Xu, and Yuanchun Shi. Reflectrack: Enabling 3d acoustic position tracking using commodity dual-microphone smartphones. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pages 1050–1062, 2021.

[71] Yiwei Zhuo, Hongzi Zhu, Hua Xue, and Shan Chang. Perceiving accurate csi phases with commodity wifi devices. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017.