ITrackU: Tracking a Pen-like Instrument via UWB-IMU Fusion

Yifeng Cao
ycao361@gatech.edu
Georgia Institute of Technology

Ashutosh Dhekne
dhekne@gatech.edu
Georgia Institute of Technology

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

ABSTRACT

High-precision tracking of a pen-like instrument's movements is desirable in a wide range of fields spanning education, robotics, and art, to name a few. The key challenge in doing so stems from the impracticality of embedding electronics in the tip of such instruments (a pen, marker, scalpel, etc.) as well as the difficulties in instrumenting the surface that it works on. In this paper, we present ITrackU, a movement digitization system that does not require modifications to the surface or the tracked instrument's tip. ITrackU fuses locations obtained using ultra-wideband radios (UWB), with an inertial and magnetic unit (IMU) and a pressure sensor, yielding multi-dimensional improvements in accuracy, range, cost, and robustness, over existing works. ITrackU embeds a micro-transmitter at the base of a pen which creates a trackable beacon, that is localized from the corners of a writing surface. Fused with inertial motion sensor and a pressure sensor, ITrackU enables accurate tracking. Our prototype of ITrackU covers a large $2.5m \times 2m$ area, while obtaining around $2.9mm$ median error. We demonstrate the accuracy of our system by drawing numerous shapes and characters on a whiteboard, and compare them against a touchscreen and a camera-based ground-truthing system. Finally, the produced stream of digitized data is minuscule in volume, when compared with a video of the whiteboard, which saves both network bandwidth and storage space.

CCS CONCEPTS

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

ACM Reference Format:
Yifeng Cao, Ashutosh Dhekne, and Mostafa Ammar. 2021. ITrackU: Tracking a Pen-like Instrument via UWB-IMU Fusion. In *The 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '21), June 24–July 2, 2021, Virtual, WI, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3458864.3467885

1 INTRODUCTION

This paper presents ITrackU, a system that enables accurate movement-tracking of a pen-like instrument across a large surface. Our goal is to continuously capture the instrument's location in order to allow replication, replay, or digitization of the actions performed. We aim to do this without embedding electronics into the work-surface or in the tip of the pen-like instrument, thereby resulting in a system that costs a fraction of the current smart-surfaces. Such technology has widespread applications, for example, in education (enabling an instructor to write on a large whiteboard while automatically digitizing the writing in real-time), in industrial IoT and robotics (monitoring the actions of miniature robotic arms assembling electronics), in artistic works (capturing the strokes in calligraphy), and so on. A common thread binds these rather diverse applications—it is impractical to embed electronics into the work-surface (because that modifies the surface), or the tip of the work-instrument. This paper reports on our efforts to design, prototype, and test a system that captures vital data about the instrument's location, its tilt, and the exact moments when it touches the surface. To achieve this, we design and deploy a novel sensor fusion approach that combines precise ultra-wideband wireless localization, inertial sensors, and pressure sensors. ITrackU generates a stream of touch locations similar to what a touch-screen would produce. By recording just the relative locations, ITrackU stores a minimal set of data about the actions on the surface. ITrackU represents a cost-effective, low-bandwidth solution to the problem of tracking a pen-like instrument.

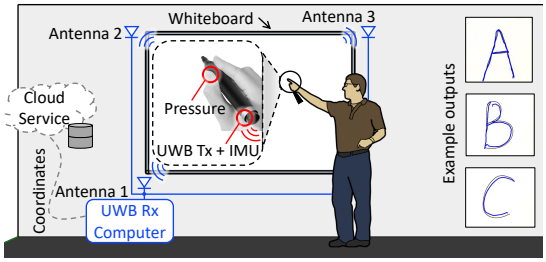

Figure 1: Example ITrackU use for online education.

As an example to demonstrate the capabilities of our system, we have chosen the educational context as a representative case of the many applications mentioned above. Whereas this paper's evaluation compares the data produced by ITrackU with the writings and drawings of shapes on a whiteboard, ITrackU's design is generic, and in principle, can be applied to other contexts as well. To this end, we have instrumented a whiteboard marker pen (see Fig. 1) and treat the writings on the whiteboard as ground-truth.

The key capability we are seeking to develop is low-cost localization of a pen-like instrument's tip on a work surface. However, obtaining the tip's location without deterring from the instrument's functionality, and without altering the work-surface, needs a novel approach. The primary challenges in accomplishing this goal are:

(1) How to perform millimeter scale localization over a surface larger than 2 meters in length?
(2) Since the tip should not be altered, how to infer the tip's location if a different part of the pen is localized?

MobiSys '21, June 24–July 2, 2021, Virtual, WI, USA

ACM ISBN 978-1-4503-8443-8/21/06…$15.00
https://doi.org/10.1145/3458864.3467885

(3) How can we detect if the pen's tip is actually touching without modifying either the surface or the tip?
(4) Can the hardware requirements be kept low to develop a low-cost solution (low deployment costs)?
(5) How to minimize data capture thereby reducing compute, storage, and transmission requirements (low operational costs)?

The core technique in ITrackU is to localize the pen's base (instead of the tip), and estimate the tilt angle and instances when the pen touches the surface to approximate the locations of the tip as it moves on the surface. Of course, fine-grained localization is, in itself, an open problem, irrespective of whether we localize the pen's tip or the base. A significant part of ITrackU's design is focused on exploring such fine-grained wireless localization. Our approach in this work is to perform active localization of a tiny ultra-wideband (UWB) transmitter module on the pen's base. This transmitter continuously emits UWB wireless packets that are captured by a *single* multi-antenna UWB receiver (keeping the hardware cost low) situated outside the work-surface. The antennas are located far apart, at different corners of the work surface (see Fig. 1). The time and phase differences between the signal received by each of the corner antennas are analyzed to track even minute changes in the pen's location. Now, to translate the pen-base's location to the tip's location, we embed an inertial and magnetic unit (IMU) that determines the pen's tilt angle. The IMU data, when fused with UWB localization also helps in improving localization accuracy during instances when UWB signals are obstructed. Finally, to detect the instances when the pen touches the surface, a pressure sensor is attached to the *side of the pen*, which helps measure the pressure exerted by the user's grip that inadvertently increases due to the friction between the pen and the surface—this force is markedly significant when the pen touches the surface. The IMU and pressure sensor data is aggregated at the pen's transmitter module and encoded into successive UWB packets. Thus, the multi-antenna receiver module can simultaneously analyze the UWB signal's time-difference-of-arrival at the corner antennas (to localize the base of the pen), and, the tilt angle and touch instances. The final output from the receiver module is just a stream of touch locations; a rather minimal amount of information which can be easily stored or transmitted over the Internet to client devices where the user's writings can be recreated in real-time.

Whereas we are not the first to show the possibility of digitizing a writing surface through localization, existing solutions [53, 55, 63] have focused only on small surfaces. Our choice of UWB allows much larger surfaces (our evaluation is over a $2.5\,m \times 2\,m$ area) to be digitized in a variety of different environments. We do require the pen (or other similar instruments) to be embedded with a wireless transmitter module and this can be viewed as a shortcoming. However, we argue that the accuracy requirements of our applications demands precise localization of the pen's tip without much room for error, which is extremely difficult using passive techniques. Therefore, we report $90\%tile$ errors in most of our evaluation. Our **main contributions** are as follows:

(1) A precise ultra-wideband localization system using a single muti-antenna UWB receiver outside the work surface to capture the pen-like instrument's location.
(2) A UWB-IMU fusion algorithm to improve localization accuracy and compensate for the pen's tilt to accurately localize the pen-tip's location.
(3) A grasp pressure sensor based method to accurately detect instances when the pen touches the surface.

In our evaluation, accurate ground truth is obtained using a camera-based system for visual inspection as well as a touch-screen based system for instantaneous comparisons of the produced location streams. We show that the median location error is $2.9mm$ and the $90\%tile$ error is $7mm$. Fig. 1 shows a sample of characters drawn using ITrackU overlayed on photos of the actual trace. Our micro-benchmarks show the utility and importance of IMU in correcting UWB errors, and the effectiveness of using the pressure sensor. Finally, the minuscule volume of data produced is compared with a video of the writing showing almost 200× reduction in storage and transmission bandwidth requirements.

2 RELATED WORK

Radio front-end: High-precision RF-based tracking has been extensively explored. Position can be inferred from wireless signature like received signal strength (RSS) [26, 32, 50] and phase [9, 10, 39, 53, 54, 57, 58]. However, pure RSS-based approaches have poor accuracy as large as $1m$ [32]. RF solutions to high-precision tracking include [29, 51, 53, 55, 58]. RF-IDraw [51] tracks motion with the interferometry property of multiple RFIDs and achieves 3.7cm median error within 5m range. However, the high cost of a RF reader and its numerous antennas makes this solution less desirable for users. PolarDraw [43] reduces the required number of antennas for RFID tracking by inferring the orientation and position of the RFID-tagged pen through electromagnetic polarization features; indeed an intriguing approach. mTrack [53] is the first sub-centimeter solution, which tracks the phase change of reflected signal from the object using 60GHz mmwave radios. MilliBack [55] embeds a backscatter tag in the pen and tracks its position with phase differential iterative schemes, achieving a median error of $4.9mm$.

UWB strikes a good trade-off between precision and propagation range, resulting in abundant applications in localization [13, 16, 20, 30]. Two way ranging (TWR) is the most widely used method for UWB to achieve centemeter level accuracy [1]. ITrackU takes a different approach than TWR. We fuse IMU with UWB, resulting in a robust *mm* level accuracy.

Acoustic: Acoustic signal has been emerging as a rising technique for tracking [8, 12, 28, 35, 37, 52, 60–62]. Strata [63] achieves $8mm$ error by tracking the position from the phase shift of the audio signal. However, the limited propagation range, an inherent weakness of audio signals, limits the widespread use of acoustic-based tracking [11].

IMU: IMU sensors can be used for tracking by double integrating the measured acceleration. However, because of the drifting nature of double integration [49], IMU sensing is typically applied in scenarios allowing frequent recalibration. For example, ArmTrak [44] and ArmTroi [33] tracks the motion of arms, which can be recalibrated by the fixed structure of the human skeleton.

Heterogeneous approaches: Jointly using multiple streams of data, such as the audio-vision signal [45, 64] or RF-IMU data [18, 19], in one application provides the flexibility of synthesizing benefits from different sensors. For instance, iBall [18] combines UWB and

Approach	Type	Error	Cost	Range (m)
RF-IDraw [51]	RFID	37mm	High	$\sim 5 \times 6$
mTrack [53]	60GHz	8mm	High	1×1
PolarDraw [43]	RFID	103mm	Middle	$\sim 0.56 \times 0.865$
Strata [63]	Acoustic	8mm	Low	0.5×0.5
MilliBack [55]	Backscatter	4.9mm	High	0.5×0.8
ITrackU	**UWB+IMU**	**7mm**	**Low**	**2.5×2**

Table 1: High-precision tracking approaches.

IMU data to track the motion of a cricket ball, in which UWB is used for localization and IMU estimates spin.

Commercial products: Current mainstream products on the market [5–7] are mostly surface assisted and based on capacitive sensing or resistive sensing of a stylus. The main drawback of these solutions is that they need support from specialized screen or surface hardware—the cost increases dramatically with the size of the screen and these mechanisms cannot provide a generic solution for use-cases where the interaction surface cannot be modified. MimioCapture ink recorder [14] is an interesting product that recognizes these shortcomings of existing approaches. It offers a solution very close to ours without relying on smart-screens. Instead, it uses infrared and ultrasound to localize a pen. MimioCapture provides a pen-sleeve[1] which can be loaded with standard dry-erase markers. However, the MimioCapture kit is quite expensive, with $696 for the main capturing device and $285 for the pen-sleeve (discounted academic price). Additionally, the ultrasound transmitter in the sleeve generates audible frequency noise due to ultrasound hardware non-linearities. This property degrades user experience; (i) the teacher can clearly hear the emitted sound since the pen is relatively close to the teacher's ears when writing, (ii) these audible fallouts can be captured and amplified by nearby microphones or hearing aids [41, 42], (iii) prolonged exposure to ultrasound is undesirable [27]. In contrast, ITrackU uses UWB radio transmissions and IMU for user-imperceptible localization at a lower cost.

Some closely related research works are compared in Table 1. ITrackU provides benefits in multiple dimensions, including precision, range, cost, on tracking through fusion of UWB and IMU. Such fusion takes the advantages of both UWB and IMU, achieving a median error of 2.9mm (7mm at 90%) in a range of 2.5$m \times 2m$ meanwhile retaining robustness even in some extreme cases.

3 SYSTEM OVERVIEW

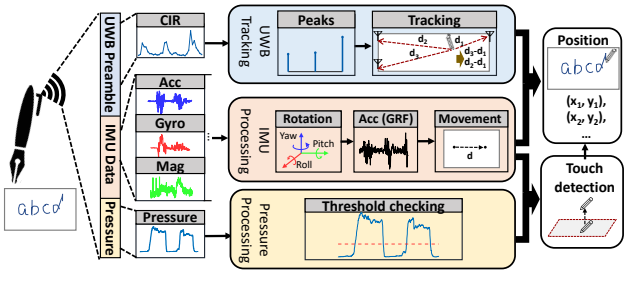

Figure 2: Processing pipeline of ITrackU.

[1]Mimio also provides an inkless product with a special stylus, but we focus on its inked version since it comes close to ITrackU's functionality.

The core idea of ITrackU is the fusion of wireless localization, inertial sensor based orientation and tracking, and touch detection using a pressure sensor, to create a system that produces an accurate stream of coordinates capturing the locations where a pen touches the surface. Our fusion approach compensates for each individual sensor's shortcomings. This carves out a unique place for ITrackU in the multi-dimensional trade-off of accuracy, range, and cost. Fig. 2 shows the various modules that make up the ITrackU system. Here, we give an overview of the modules' interactions, and in Section 4 we delve deeper into each module's functioning.

We refer the reader to Fig. 1 for clarity of the following explanation. A custom-made electronic cap is attached to the base of a pen-like instrument to be tracked. In our case-study specifically, we attach the cap to the base of a marker pen. This cap houses a micro-controller that records data from an on-board inertial sensor and a pressure sensor. A pressure sensitive strip runs down the pen's side to a point where the user holds the pen. It captures the force exerted by the user to detect touch events. The cap also houses a UWB wireless transmitter which periodically sends UWB packets containing this IMU and pressure sensor data. These packets are received by a UWB receiver with 3 antennas. The UWB packets here perform double-duty; they carry sensor data from the pen to the UWB receiver, and also inherently expose timing information which the receiver uses to localize the pen.

The core innovation that allows ITrackU to *localize* the pen from this unidirectional stream of UWB packets, is a 3-antenna setup, with each antenna placed at a different corner around the work surface. All antennas are connected to a single UWB receiver using different lengths of RF cable (observe the periphery of the whiteboard in Fig. 1). This results in multiple copies of the transmitted signal arriving at the UWB-receiver delayed by a few nanoseconds. Given UWB's large bandwidth (1 GHz), it is possible to identify subtle timing differences between signal copies by studying the channel's impulse response (CIR). The radio's physical layer (PHY) obtains the CIR by correlating the received preamble with known golden sequences. Analyzing reception timings and signal phases in the CIR allows localization of the pen's transmitter.

However, just obtaining UWB CIR is insufficient to localize the pen on the surface. There are 2 issues unresolved: (1) the UWB-transmitter is actually attached at the base of the pen, while we need the tip's location, and (2) UWB CIR may not be always pristine; it might occasionally get polluted by occlusions or multipath. To solve the **first issue**, we need to know additionally the posture of the pen (assuming we know its length). The IMU information inside every UWB packet provides 9-degrees of freedom (DoF) information about the pen's inertial frame of reference which allow computation of the pen's orientation. The **second issue**—occasional UWB signal blockage—intermittently causes poor localization accuracy. Phase based tracking can amplify such intermittent errors due to sudden phase wraps, ITrackU depends on the IMU to provide a second stream of location estimates derived from integrating the pen's movements. Erroneous changes in signal phase occurring due to UWB blockage can be corrected by independent information from IMU sensors.

Fusing information from the UWB sensor and the IMU sensor can offer precise localization of the pen's tip. However, the position difference between the tip *touching* the surface or *hovering* just

above it, is too subtle for localization to distinguish. Given the limitation that neither the touched surface, nor the pen's tip can be instrumented, we have crafted a novel way to detect touch—detect the force exerted by the user's grip on the pen, instead. When a pen's tip touches a surface, the friction between the surface and the pen increases the reaction force exerted by the pen on the user's grip (and vice-versa). We measure this using a pressure sensor and report precise timings when it crosses a threshold. Thus, the location information is filtered based on the touch information produced by the pressure sensor. Finally, a stream of (x, y) coordinates are produced by the corner UWB-receiver. These coordinates can be stored or transmitted over the Internet for rendering the user's drawings on remote computers.

4 SYSTEM DESIGN

We now describe the system design in detail, starting with the UWB tracking module, followed by IMU fusion, and finally by the touch detection module.

4.1 Tracking with UWB

Ultra-wideband radios are frequently used for localization and ranging. Typically UWB localization protocols require a two-way message exchange between a device pair [4, 22] for appropriate clock offset and clock drift compensation. However, each message exchange consumes time, therefore a multi-message protocol reduces the distance computation update rate. We break away from this multi-message requirement, to improve update rate, and instead rely only on a unidirectional flow of messages from the pen's base to a UWB-receiver. Using a unidirectional flow for localization is only possible when multiple receivers are time-synchronized. In such cases, a (reverse) time-difference-of-arrival (TDoA) protocol computes the *difference* in the times at which a mobile transmitter's signals arrive at *three synchronized receivers* (at a minimum) [24]. One of the core innovation of this work is in capturing UWB messages sent by a transmitter, using a *single UWB receiver*. This is feasible in our use-case since the pen is expected to move within a rather confined space, and we can place three antennas at three different corners of the work-surface, while being connected to a single UWB-receiver. Fig. 3 shows the processing at the UWB receiver, where the TDoA between every antenna-pair produces a hyperbolic locus of the pen's potential locations. Intersection of the three hyperbolas, each obtained from the time-difference between a unique pair of antennas, solves for the pen's exact location. Note that simply solving the equation will lead to ambiguity with two position candidates. However, since UWB can provide high-resolution CIR, we can uniquely determine which of the air-paths is longer, and solve for locations within the expected whiteboard space, thereby removing ambiguity. Each antenna is connected to the receiver using a different RF cable length and power attenuators—this ensures each antenna's signal copy is time-separated on the obtained CIR,

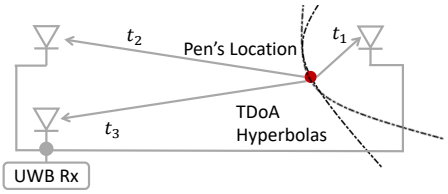

Figure 3: TDoA based solving of intersecting hyperbolas.

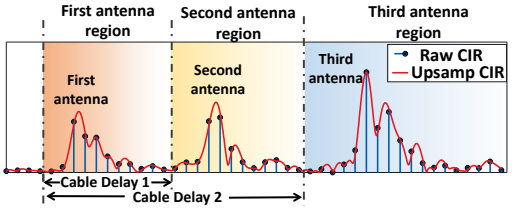

Figure 4: The CIR received by 3 antennas of the receiver.

and yet has sufficient power to be observable. The longer the RF cable, more is the attenuation, which we compensate for, by adding extra attenuators on the shorter RF cables.

The above foundational discussion about solving hyperbolas for localization still ignores certain details. For example, the time UWB signals spend on the long RF cables connecting the antennas is coupled with that spent in air. We are interested in the time spent on the air paths only, which we call air path differential (APD), which we discuss next.

■ Calculating Air Path Differential (APD)

Our goal is to extract air path time differentials from the CIRs obtained at the UWB receiver. Mathematically, the baseband demodulated received signal $y(t)$ comprises of multiple attenuated (a_k) and delayed (τ_k) copies of the transmitted signal $x(t)$ [48].

$$y(t) = \sum_k a_k(t) x(t - \tau_k) e^{-j2\pi f_c \tau_k}. \tag{1}$$

In ITrackU, we have purposefully created a 3-antenna setup that adds a per-antenna delay on received signals. For the i^{th} antenna, the total delay observed by the receiver, τ_i has two components: the propagation delay in the RF cable $(\tau_{c,i})$, and that in air $(\tau_{a,i})$:

$$y(t) = x(t) e^{-j(2\pi f_c (t - \tau_{a,i} - \tau_{c,i}) + \phi_0)} \tag{2}$$

where, ϕ_0 is the receiver's arbitrary phase offset at the time of packet reception. A delay implies that the wireless signal traveled an extra distance corresponding to that delay. For a signal with wavelength λ, a delay τ_i corresponds to n_i integer wavelengths and a residual $\phi_i/2\pi$ fraction of λ. Since the reported CIR is a complex number the phase is extractable, and quite stable (more about the stability later). Therefore, expanding eq. (2) to include phase and integer wavelengths, in the cable (subscript c), and the air (a) we obtain the following expression:

$$y_i(t) = x(t) e^{-j2\pi f_c \left(t - (n_{c,i} + \frac{\phi_{c,i}}{2\pi})\frac{\lambda}{c} - (n_{a,i} + \frac{\phi_{a,i}}{2\pi})\frac{\lambda}{c} + \phi_0\right)}, \tag{3}$$

where, c is the speed of light. This shows the **two hindrances** to extracting the APD: (i) every packet has an arbitrary phase ϕ_0, and (ii) signal travelling through the cable's length pollutes the air path.

To solve the above two hindrances we introduce a double differential approach. First, observe that eq. (3) written out for all antennas share identical phase offset ϕ_0, we subtract the various antenna equations from each other, and eliminate the ϕ_0 term. Writing just the relative delay $D_{r,i}$ equations for antenna i and antenna 1 we get:

$$\begin{aligned} D_{r,i} = &\frac{\lambda}{c}\left((n_{c,i} + \frac{\phi_{c,i}}{2\pi}) + (n_{a,i} + \frac{\phi_{a,i}}{2\pi})\right) \\ &- \frac{\lambda}{c}\left((n_{c,1} + \frac{\phi_{c,1}}{2\pi}) + (n_{a,1} + \frac{\phi_{a,1}}{2\pi})\right). \end{aligned} \tag{4}$$

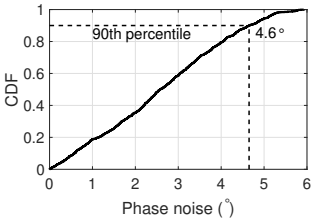

Figure 5: The relative phase of UWB deviates little.

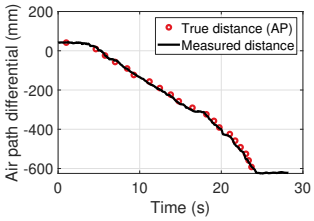

Figure 6: Anchor points' true phase vs. measured phase.

Second, observe that the cable-delay remains constant over time for any antenna. Therefore, differencing consecutive packets received over the same antenna cancels out the cable delay. Performing this difference over the relative delays from eq. (4), we obtain:

$$
\begin{aligned}
{}_{\Delta}D^{j}_{r,i} = \frac{\lambda}{c}\left[\left(n^{j+1}_{a,i} + \frac{\phi^{j+1}_{a,i}}{2\pi}\right) - \left(n^{j+1}_{a,1} + \frac{\phi^{j+1}_{a,1}}{2\pi}\right)\right] \\
-\frac{\lambda}{c}\left[\left(n^{j}_{a,i} + \frac{\phi^{j}_{a,i}}{2\pi}\right) - \left(n^{j}_{a,1} + \frac{\phi^{j}_{a,1}}{2\pi}\right)\right].
\end{aligned} \tag{5}
$$

This difference in phases over time has only air-path terms which maps to the change in location of the pen (APD). Suppose the distance of the transmitter from the i^{th} antenna during UWB packet j is d^{j}_{i}. Then change in the APD can be written as:

$$
{}_{\Delta}APD = (d^{j+1}_{i} - d^{j+1}_{1}) - (d^{j}_{i} - d^{j}_{1}) =_{\Delta} D^{j}_{r,i}. \tag{6}
$$

Additional simplification of ${}_{\Delta}D^{j}_{r,i}$ is possible if the relative movement between consecutive packets is within half a wavelength. Then, $n^{j+1}_{a,i} = n^{j}_{a,i}$ and the relative APD is calculable just using phases:

$$
{}_{\Delta}\phi^{j}_{r,i} = \phi^{j+1}_{r,i} - \phi^{j}_{r,i}, \tag{7}
$$

where $\phi^{j}_{r,i}$ is the relative phase between antenna 1 and antenna i for the j^{th} UWB packet. Of course, if during consecutive UWB packets the transmitter has moved more than one wavelength, then $n^{j+1}_{a,i} \neq n^{j}_{a,i}$ and eq. (5) will produce a phase-wrap error. We will discuss these later in the context of inertial sensor fusion.

Equation (7) is used as a foundational mechanism to localize the pen in ITrackU. One aspect that we alluded to earlier still needs to be inspected: if we are relying heavily on phases, we must first ensure that phase is indeed reliable. To this end, we obtain empirical data by keeping the pen stationary and recording the changes in relative phase difference between two antennas. Fig. 5 shows the change in relative phase over 937 independent UWB packets. We observe that the 90%$tile$ range of phase[2] is about 4.6°. For UWB center frequency of 3.993GHz, the distance precision is $\frac{4.6}{360} \cdot \frac{3\times10^{8}}{3.993\times10^{9}} \approx 0.96mm$, giving the practical limit of ITrackU. Note that phase precision exceeds time precision by a couple of orders of magnitude—compare the 1 mm phase precision with about 15 cm time precision in Fig. 8.

To verify that eq. (7) produces results close to ground truth measurements, we setup a simple experiment with two fixed receiving antennas placed 3m apart. We move a transmitter along a marked path for a distance of $\approx 50cm$. The path has 20 marked points with known positions. A camera records timestamps when the transmitter passes over the markers. Fig. 6 shows that the estimated APD from the phase measurements of eq. (7) matches ground truth well.

[2]This includes the hardware phase noise.

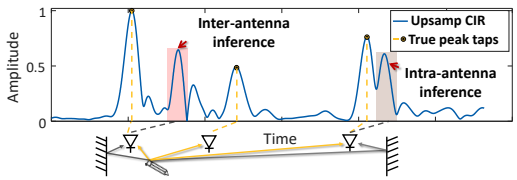

Figure 7: Inter-antenna interference and intra-antenna interference mislead the tap selection process.

■ Extracting phase from CIR

We obtain a raw CIR with samples every 1 ns (1 GHz sampling rate) from the UWB hardware. Each sample in a CIR is also called a tap. Fig. 4 shows the raw CIR taps obtained at the receiver with three prominent pulses, corresponding to received signal over each antenna. However, even with nanosecond resolution in time, the exact moment when the signal peaks can be missed as the true peak frequently lies between two sampled taps (e.g., the peak tap in the first impulse in Fig. 4). To address this problem, we upsample the raw CIR to 64GHz (15.625 ps duration between taps). This sampling frequency is sufficient to capture the "missing" peak between two CIR taps, as shown in Fig. 4. ITrackU then finds the peak taps in each CIR and tracks the phase, followed by APD calculations.

■ Tap selection despite multipath

Tap selection issues: The accuracy of tracking depends on accurate peak-detection in each impulse. Rich multipath in an indoor environment breaks the simplicity of picking the peak with the highest amplitude. Fig. 7 shows examples of how multipath can influence the tap selection. There are 2 primary problems:

(1) Inter-antenna inference: The delayed environmental reflections of a prior antenna's signal can confuse the selection of the subsequent antenna's signal.
(2) Intra-antenna inference: The peak in an impulse is polluted by nearby multipath received by the same antenna (an object close to the antenna).

These problems cause significant errors in phase measurements. Existing works, such as Strata [63], reduce the search-space by filtering out the taps whose corresponding physical position lies outside the field; however, this works only for small fields. In our $2.5m \times 2m$ area, many locations can have ambiguous tap selections even after search space reduction. Furthermore, in a multipath-rich indoor environment, multipath taps can occur anywhere in the CIR. Reducing search space cannot filter out multipath.

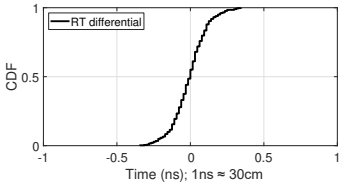

Figure 8: RT differentials.

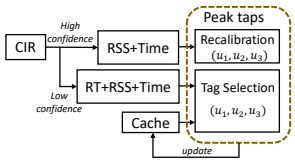

Figure 9: Tap selection.

Solution: We take an architectural and an algorithmic approach for accurate tap selection. First, to mitigate confusion caused by inter-antenna interference, a careful choice of delaying cables and RF attenuators is important. The insight is that if a multipath arrives later than a certain time threshold T_{max} after the direct path, it is likely to have much lower RSS. With this assumption, we use

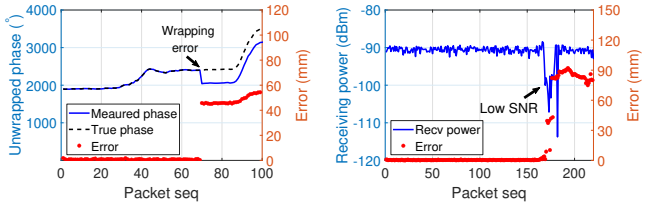

Figure 10: The effect of (a) phase wrapping, (b) blockage.

delaying cables to ensure the direct-path impulses in different antennas are separated by at least T_{max}. Then we fine-tune the value of RF attenuators placed at each antenna to limit the signal power received by earlier antennas ensuring the signal from a posterior antenna has larger amplitude. Thus, the multipath from the first antenna will always be weaker compared to the peak of the direct impulse of the second antenna, and so on. The true peak can be discriminated from a multipath tap by simple amplitude check.

Intra-antenna interference, however, cannot be solved as above because the interfered signals are received by the same antenna. Instead, we exploit the observation that ***the relative time (RT) of peak taps change little in two adjacent UWB packets*** (If the peak tap of the first impulse is t_1 and the peak tap of the second impulse is t_2, their relative time is $t_2 - t_1$). To ground this hypothesis in empirical results, we moved a transmitter at a normal writing speed and recorded the RT differential between adjacent packets. We observe from Fig. 8 that the RT differentials are limited in $\pm 0.5ns$. In contrast, the multipath tap is typically separated from the direct path tap by at least $2ns$. Therefore, if we already know the RT in a CIR packet is τ, and we have detected that the first peak tap of the next CIR is t_1, then we need to only search taps lying in the range of $(t_1 + \tau - 0.5, t_1 + \tau + 0.5)ns$. As a procedure, we first extract all peaks in the CIR and filter out low-amplitude peaks. From the remaining peaks, we compare the RT of each peak in this CIR with the RT of peak taps cached from previous packets. We choose the 3-tuple that satisfies the above timing requirement, keeping only the one with a largest amplitude and smaller timestamp. Of course, if we keep using RT differential for tap selection we may propagate errors, whereby the error in the peak tap selection in one CIR will also influence the peak tap determination of next CIR. To avoid it, we constantly recalibrate the RT when we encounter *high-confidence CIR* in which peak taps are clean and prominent. Fig. 9 shows this complete tap selection pipeline.

■ Limitations of using only UWB localization

The above phase-based tracking scheme seemingly gives a complete solution to a low-error tracking system. However, there are two unsolved fundamental problems, which exist widely in phase-based wireless tracking system. We specify these two problems in this section and explain how they erode tracking accuracy.

Wrapping error: As mentioned before, the receiver needs to extract the peak tap's phase in the CIR. Since the phase value is bounded in $[-\pi, \pi]$, there is no way to detect phase wraps, just from the phase. This problem is known as *integer ambiguity*. Our system, as well as several other existing works, track the phase change between every successive packet to counteract the effect of integer ambiguity. This works so long as the phase change between two packets never exceeds half a cycle (π in phase), which sets an upper bound on the object's moving speed. Quantitatively, given the average time gap between packets t_{gap}, the allowed maximum moving speed is $\frac{\lambda}{2t_{gap}}$. For UWB with center frequency 3.993GHz, with a 50Hz packet transmission rate, the maximum speed is $\frac{0.0751}{2\times 0.01} = 1.875m/s$. However, human writing can reach an instantaneous velocity as high as $3m/s$, which means that wrapping errors are feasible. Fig. 10(a) shows the effect of just one wrapping error—the APD suddenly deviates by at least one wavelength, leading to centimeter or even decimeter errors.

Low-quality CIR: Wireless signals are vulnerable to environmental factors such as human motion and blockage, both of which can produce low-quality CIR. The receiver cannot extract the effective phase from these CIR and therefore, leaves a void period during which tracking is infeasible. Note that the effect of the void period is cumulative, meaning all the subsequent errors add to existing errors. This effect can be seen from an experiment where we fixed the transmitter and receivers and the researchers walked through the experimental area. The receiver tracks the motion of the transmitter (should be 0 since the transmitter is stationary). Fig. 10(b) shows the correlation of the variations in the received signal strength and the absolute position error.

Given these UWB tracking issues, we ask: *Can we have a complementary scheme which assists UWB retain robustness, mitigating effects of both phase-wrapping errors and occasional low-quality CIR?*

4.2 Augmenting Location Tracking with IMU

To maintain robustness of localization accuracy even when UWB wireless signals face the problems of wrapping and occlusions, we require a fundamentally different technology which will perform tracking without emitting signals. An inertial sensor presents promising possibilities. It measures acceleration, angular velocity, and magnetic field experienced by the pen, which helps create a deadreckoning system based on the pen's *motion vectors*. Despite wide use of IMUs in localization [17, 25, 38, 56], applying IMU in high-precision tracking tasks is not sufficiently explored—the drifting nature of IMU makes it extremely hard to provide stable low-error position measurement over a long period. Therefore, to effectively use IMU, we must first understand the merits and demerits of IMU, and then explore how it can assist in our case.

■ Background of IMU and tradeoffs vs UWB

In contrast to UWB's phase-based tracking, IMU captures the effects of movements on its inertial frame of reference. At a high level, one can obtain linear acceleration in the global frame of reference (GRF) from a 9-degrees of freedom[3] inertial and magnetic sensor and perform double integration to obtain the distance an object has moved. It might then seem that IMUs can provide us exactly what we need. However, there are pitfalls in relying on IMU alone. Table 2 lists the major differences between UWB and IMU in a tracking task. Unlike UWB, IMU is a standalone sensor—it does not need a transmitter-receiver pair to work. This property provides independence from wireless interference and blocking, and is an advantage over any wireless-based localization. Furthermore, due to the time required to extract CIR, the IMU sensor typically has a higher sampling rate ($> 200Hz$) compared to UWB ($< 100Hz$), particularly when compared with UWB's CIR processing. That means the IMU sensor can capture an object's motion over a period as short as $5ms$, yielding the potential for tracking with low error.

[3] Accelerometer, Gyroscope, Magnetometer (3-axis each).

Attribute	UWB	IMU
Sensors Needed	TX-RX pair (2)	1
RF Occlusion	High impact	Almost no effect
Sample rate	$20Hz \sim 100Hz$	$\sim 200Hz$
Precision	$< 1cm$ (ITrackU)	$t < 1s$: 90% $0 \sim 3cm$ $t \geq 1s$: Drifting error

Table 2: The comparison of properties of UWB and IMU.

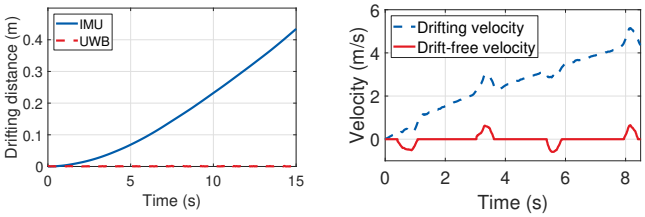

Figure 11: (a) The drift of UWB and IMU. (b) The velocity before and after drift correction.

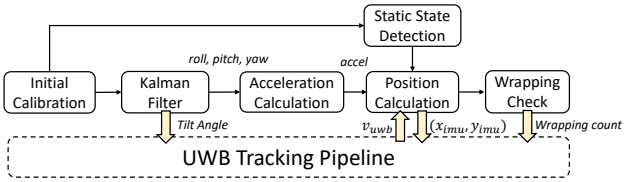

Figure 12: The UWB-IMU fusion procedure.

Despite the independence from wireless blocking and the high sampling rate, a major problem with IMUs is that they drift over time. This drifting is fundamental, and results from double integration of acceleration. To understand how this drifting behavior would affect ITrackU, we tie an IMU sensor on the marker pen and fix the marker pen on the ground. We collect IMU data for 30 seconds and observe how it drifts with time (see Fig. 11(a)). It is a stark contrast, when compared with UWB's lack of drifting (dotted red line which almost coincides with the zero-line). We make two key observations: First, *IMU drifts significantly (meter-scale) even over a few seconds*. Hence, using IMU alone in high-precision tracking for long time is not realistic[4]. The second observation we make is ***IMU retains good accuracy over sub-second durations***, if well-calibrated. This property stems from the motion equation $d = v_0 t + \frac{a \cdot t^2}{2}$. When the moving duration $t < 1s$, the square operation weakens the effect of drifting, yielding high precision over a short time, while the same squaring operation amplifies drift over longer durations. Fortunately, the problems in UWB tracking—wrapping errors, and signal blocking—are short-lived. This makes the IMU a potentially well-suited complementary sensor in robust high-precision tracking, details of which are presented next.

■ Fusion algorithm

Fig. 12 presents a high-level view of ITrackU's fusion approach. We first perform a one-shot initial calibration on the IMU sensor to remove various offsets and biases within the IMU chip. Since the acceleration data after calibration is measured in the local frame of reference (LRF), we apply a Kalman filter to infer the rotation from LRF to GRF with 9DoF sensor data. To avoid the gimbal lock problem [23], we use rotation matrix $R_{l \to g}$ instead of Euler angles to represent the mapping from LRF to GRF. In addition to obtaining the acceleration vector in GRF a_g, $R_{l \to g}$ also provides the **tilt angle** which serves to compensate the projection offset of the sensor. Of course, at this time, directly using a_g to calculate the moving distances is still prone to significant drift. To address this problem, we perform a twofold recalibration scheme on motion data where we look for two type of events: (i) when the object is static, and (ii) the movements generated from UWB tracking. After recalibration, we can solve the position from IMU data, and then fuse it into UWB tracking to improve the overall performance. Details of each module in the fusion design are presented next.

Initial calibration: Initial calibration involves the independent calibration of acceleration a_{raw}, gyroscope data ω_{raw} and magnetometer data m_{raw}. For accelerometer and gyroscope, biases are introduced by sensor imperfections, while for magnetometers, external magnetic objects cause hard iron and soft iron distortion [21]. We apply linear fitting and zero-mean to remove the initial offset, and store these values for future use. We employ a standard 10-parameter calibration on the magnetometer [2]. The calibration outputs, a_l, ω_l, m_l, serve as our fusion algorithm's initial inputs.

Rotation calculation (Kalman-filter): With a 9DoF calibrated IMU data in LRF, there are two independent streams to calculate the rotation matrix $R_{l \to g}$. The **first stream** uses the global gravity vector and magnetic vector as anchors. For a static object at time t:

$$a_g = R_{l \to g} a_l(t) = g_0, m_g = R_{l \to g} m_l(t) = m_0, \tag{8}$$

where g_0 and m_0 are known as the global gravity vector (pointing downwards) and geomagnetic vector (approximately pointing north and downwards). This stream has no drift but is prone to pollution from movements and indoor magnetic noise. **Another stream** directly accumulates ω_l to obtain the rotation. While it is not polluted by linear acceleration (movements) and magnetic noise, it drifts over time. A Kalman filter strikes a good trade-off between these two streams [31, 59]. The idea is to smooth the error through weighted average of the outputs from the above two streams, and adaptively tune the weight based on the error feedback of each sensor. For instance, if the error of gravity $e_g = \big||a_l| - |g_0|\big|$ increases, the weight of accelerometer will decrease correspondingly. The tuning repeats in every measurement to improve the rotation calculation consistently. In addition to being fed into the module of acceleration calculation, $R_{l \to g}$ also generates the **pen's tilt angle**, which removes the tracking offset caused by the distance gap between the RF transmitter and the pen's tip.

Acceleration in GRF: The rotation matrix $R_{l \to g}$ maps the local acceleration back to GRF and then we remove the gravity vector to obtain clean linear acceleration in GRF $a_g^{lin} = R_{l \to g}(t) a_l(t) - g_0$.

Position in GRF: While the IMU data has already been calibrated at this time, integrating a_g^{lin} to obtain position directly still incurs drift resulting from double integration. To remove it, we apply a sliding window where the time spans less than $1s$. Picking $t < 1s$, mitigates the drift of the second term (squared) in $d = v_0 t + \frac{a \cdot t^2}{2}$, though the drift in the first term $v_0 t$ remains unsolved. The difficulty lies in that the initial velocity v_0 in every new calculation of the distance depends on the end velocity in the previous calculation; causing lasting effects beyond $1s$. To address this, we correct v_0 through a 2-phase scheme that we call *filter-and-reset*. In the *filter* phase, we apply a high pass filter [34] on a_g^{lin} to remove drift. The cutoff frequency is set to $0.2 f_s$, where f_s

[4]High-precision IMUs cost thousands of dollars each [47].

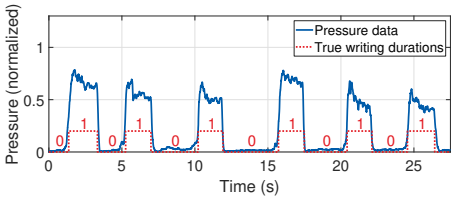

Figure 13: Surface touch detected through change in user's grip-pressure correlates well with ground truth.

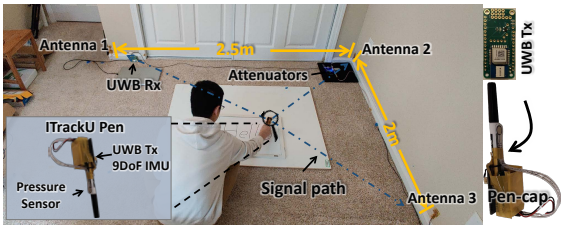

Figure 14: The ITrackU testbed and UWB pen-cap

is the IMU sampling rate. However, finding a general cutoff frequency is challenging, causing slight drifts to remain. We eliminate such drift completely in the *reset* phase. Specifically, we employ the following two simple rules for *reset*: (i) When the object is stationary, the velocity vector is $(0, 0, 0)$, (ii) When we have a high-confidence UWB CIR, the velocity vector should conform with the displacements obtained by UWB tracking. Based on the first rule, we make stationary-state inference by monitoring $(|a_l| - |g_0|)$ and $|\omega_l|$, which should be 0 when the object is stationary (or moving at a constant speed, but that rarely occurs in our use-cases). Reset is triggered when the UWB signal has high confidence. Based on the position provided by high-confidence UWB signal, we can obtain a reference initial velocity: $v_0'(t) = \frac{d}{dt}(pos(t +_\Delta t) - pos(t))$. Since the reference velocity $v_0'(t)$ is not polluted by IMU drift, our initial velocity remains accurate for movement calculation. We jointly use the above two rules for sample-wise velocity correction. Fig. 11(b) shows the velocity before and after drift removal. By applying the *filter-and-reset* scheme, the position error in a $t = 1\,s$ sliding window can be reduced from $1\,m$ to $< 3\,cm$, and reduces to sub-centimeter when $t < 0.3\,s$.

Fusion Benefits: Fusion with IMU benefits overall tracking in a couple of ways, which answer the question posed at the end of Section 4.1. First, fusion addresses the wrapping error by estimating the APD between two packets with IMU data. Recall that wrapping error occurs when the APD between two successive packets exceeds $\frac{\lambda}{2}$. UWB CIR is just incorrect about the count of integer wraps while still providing accurate phase measurement. IMU data fills the gap. For instance, if we find the APD calculated from IMU data lies in $[\frac{\lambda}{2}, \frac{3\lambda}{2}]$, the count of wrapping cycles should increase by 1. Such wrapping check happens for every UWB packet. With the correct wrapping cycle estimation, together with high precision tracking using phase, we can retain accuracy similar to the no integer ambiguity case.

Second, fusion provides a "safety net" on the tracking system, especially for short-lived low quality CIR caused by temporary occlusion events. A UWB-only tracking scheme will experience significant accuracy degradation during this time, and might sustain even after the event. ITrackU fills the gap by using IMU data instead. By positioning with IMU, we can achieve $< 3cm$ errors for about $1\,s$. Once the UWB signal quality recovers, it takes over, without any error propagation.

4.3 Touch Detection

The last leg that completes the tracking is the touch event detection. The fundamental question we are asking here is: *when is the pen touching the surface?* Without modifying the tip of the pen, and keeping the surface as it is (no modifications are made to the surface for touch detection), this is a difficult question to answer. One approach towards touch detection is to extract signal features as the input of a decision model, like [46, 53], or to detect the z-axis location of the pen. However, the performance of signal-feature based inference is tightly correlated with the quality of received signal, yielding high false-positive or negatives in low-SNR conditions. Likewise, we would need sub-mm precision in the z-axis to detect such subtle movements of the pen. Instead, viewing this problem from the sensor fusion angle we try to achieve touch screen-like accuracy. The core intuition in solving the touch detection problem is that users have to hold the pen more firmly when writing. If we embed a pressure sensor on the side of the pen, the real-time pressure change, accompanied by the tilt angle relative to the reference frame of the writing plane, can solve this problem more generically by a simple pressure-threshold check. To verify this, we attach a force sensitive resistor on the side of a pen and track the pressure when writing. A resistive touch-screen is used to detect the exact ground-truth instant when the surface is touched and the detected touch events are compared. Fig. 13 shows the high correlation between the pressure data and the touch-screen's output. This correlation is generic since additional pressure is always required to overcome the friction of the surface when writing. Pressure is automatically relieved when the friction is absent (when the user lifts the pen).

5 IMPLEMENTATION

Fig. 14 shows our prototype and experimental setup with three antennas at $2.5\,m$ and $2\,m$, describing our large work area.

Prototype: The pen's *cap-setup*, which transmits UWB packets, includes a DW1000 chip ($18), the Adafruit Feather M0 ($20), a 9DoF IMU ($18), and a Sparkfun force sensitive resistor ($21) serving as the pressure sensor. The IMU consists of an iNEMO 6DoF inertial sensor (accelerometer and gyroscope) and a LIS3MDL 3-axis magnetometer. These components are assembled on a PCB board of $4.5\,cm \times 2\,cm$. The whole transmitter prototype, created using COTS components and costing less than $80, weighs about 34 grams—lightweight enough that the pen is still usable. The cap collects data from the IMU and the pressure sensor with $200\,Hz$ data rate. A group of IMU measurements are radioed out as a batch using UWB packets. The UWB signal is at $3.993\,GHz$ center frequency and has a $1\,GHz$ bandwidth. The UWB packet rate is bottlenecked by the per-packet CIR extraction at the receiver. Since IMU has a much higher data rate than UWB, we embed 5 sets of IMU and pressure readings into one UWB packet for rate matching. On reception of the signal, the receiver triggers internal signal processing pipeline to generate CIR. Thus UWB packets function as both, a sensing mechanism to track the position, and a communication mechanism for packet transmission.

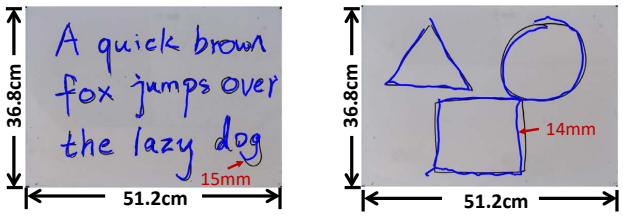

Figure 15: Examples of tracking characters and shapes. Ground truth is what the pen actually wrote on a surface.

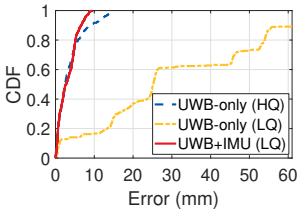

Figure 17: Benefit of ITrackU's fusion approach.

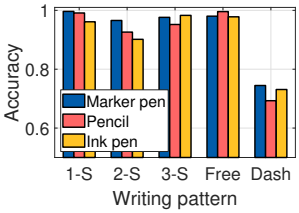

Figure 18: Touch detection: different pens and patterns.

Description	Product Number	Size
Microcontroller	ATSAMD21G18A	$7mm \times 7mm$
UWB Chip	DW1000	$6mm \times 6mm$
Accelerometer + Gyro	ISM330DHCX	$2.5 \times 3mm$
Magentometer	LIS3MDL	$2mm \times 2mm$
UWB Antenna	UWB 06	$6mm \times 8mm$

Table 3: Size of the main ICs used in ITrackU's pen attachment.

At the receiver end, three antennas are at three corners of a $2.5m \times 2m$ tracking region. They are wired to a Decawave Trek1000 development board through a 3-way RF splitter with different length of delaying cables. Cost of the receiver can be well within $100 when manufactured using integrated circuit components. In this prototype, the Decawave Trek1000 extracts a 100-tap CIR (0.4KB) and then relays the CIR and other received data over a USB-serial port to a laptop, which runs the pipeline in Matlab R2019a.

From prototype to product: One may wonder how far removed ITrackU is from a real-world pen-attachment product housed in an appropriate form-factor. Note that the main components on the pen include functional electronic components and a battery. The electronic components are a few mm^2 each (see Table 3), which makes it feasible to miniaturize the prototype into pen caps with inexpensive manufacturing technology. The size bottleneck lies in the size of the battery. Since the volume of the battery largely determines the battery's charge cycle, we perform a battery draining test with Lipo battery LP503035 (3.7V, 500mAh) and a trade-off analysis among other common Lipo batteries in Section 6.

6 EVALUATION

We evaluate the performance of ITrackU in an indoor home environment. We take a two-pronged approach to obtain the accurate ground truth. First, a $37.6cm \times 30.3cm$ resistive touch screen [3] is placed within the region on which we perform various writing tasks for evaluation—this provides a ground truth trace with a resolution of 2 *mm*. Second, for some evaluations, we use a camera based ground-truth for better accuracy. We clamp the *pen-cap-setup* on the back of a marker pen and assume the user is holding the pen as shown in Fig. 14 with the right hand. Different pen holding

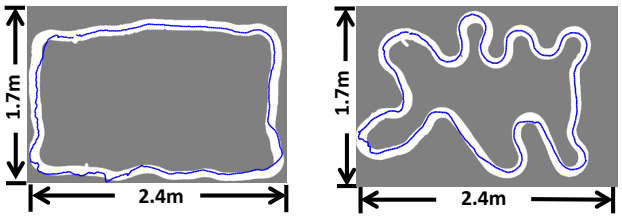

Figure 16: Examples of tracking a moving robot (blue trajectory) on a given track (white). Larger area covered.

patterns are supported so long as the signal emitted from the UWB antenna (on the pen's base) is not shielded. We also assume the user is in front of the whiteboard and that the receiver antennas are in the plane of the whiteboard, thus avoiding blocking the signal. For evaluation, we write or draw shapes on the whiteboard at normal handwriting speed, demonstrating freeform writing capability.

6.1 Overall Performance Evaluation

Visual performance: Fig. 15 visually demonstrates the accuracy of writing text or drawing a shape on a $52.1cm \times 36.8cm$ whiteboard with a maximum error of $15mm$ approximately. Demonstrating ITrackU's performance in large areas, due to difficulty in obtaining the movements' ground-truth, needs a novel approach; instead of writing, we fix the ITrackU pen on a mobile robot and let the robot move along a long toy-train track built in advance. Fig. 16 shows the performance of ITrackU in this large area.

Performance of tracking: We first evaluate ITrackU's accuracy. To measure the error quantitatively, we place the touch-screen in the experimental region providing the ground truth. Then we draw a rectangle with the marker pen equipped with all the sensors on the touch screen. We compare the performance of ITrackU in three different cases: (i) All CIRs are high-quality, and we only use UWB for tracking; (ii) Some CIRs are low-quality (possibly obstructed), and we only use UWB for tracking; (iii) Some CIRs are low quality, and we jointly use UWB and IMU for tracking. The low quality CIRs are generated by imposing Gaussian noise on the received signal. Fig. 17 shows that for high quality CIRs, using only UWB achieves a 90^{th} percentile error of $8.9mm$. The error increases drastically after we introduce some low quality CIRs (maximum error exceeds $60mm$). Such performance degradation results from phase tracking error propagation due to low-quality CIRs. However, *mm*-level accuracy is restored after we fuse UWB and IMU readings. Once the receiver detects a low-confidence CIR, the IMU performs the tracking for a short time. Benefiting from the IMU's promise of high accuracy for a short duration, ITrackU computes position with IMU, and then translates the position back to phase change. This ensures consistency of UWB phase tracking even in the presence of low-quality CIRs.

Touch Detection: Next, we evaluate the performance of touch detection with the pressure sensor. We write 5 patterns with different pen-lift and touch sequences, on the touch screen: (i) one-stroke (1-S) characters, like "C"; (ii) two-stroke (2-S) character (2-S, like "B"); (iii) three-stroke character (3-S, like "A"); (iv) free-form curve (Free); and a (v) dashed line (Dash). The touch screen records the ground-truth touch duration. Our goal is to examine the robustness of ITrackU's touch detection over increasing complexity of written artifacts. These patterns are drawn using three types of pens with

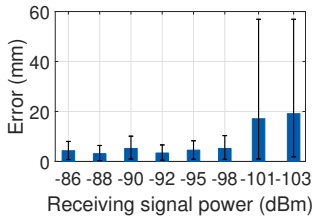

Figure 19: Error dependence on received signal strength.

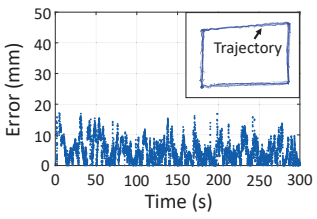

Figure 20: Temporal consistency: no drift observed.

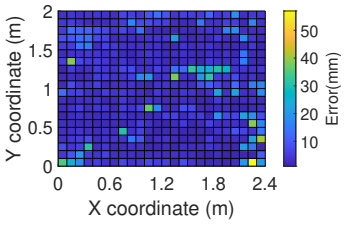

Figure 21: Spatial consistency: few poor spots.

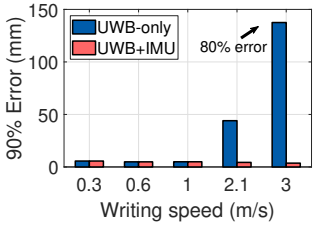

Figure 22: Writing speed: IMU fusion compensates for UWB.

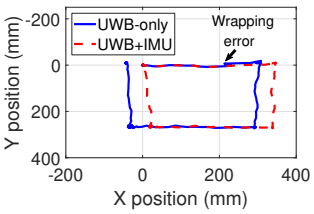

Figure 23: IMU compensates for phase-wrapping error.

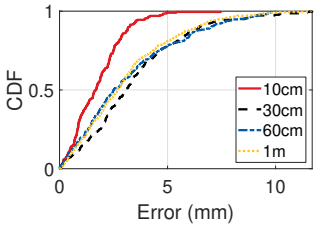

Figure 24: Sub-cm 90% errors across various fontsizes.

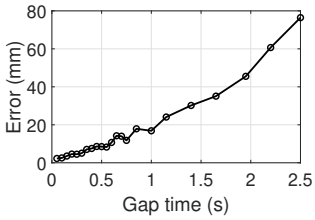

Figure 25: IMU covers for UWB's short-term absence.

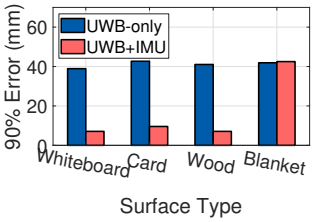

Figure 26: ITrackU's performance on different surfaces.

different tip hardness: marker-pen, pencil, and ink-pen. We detect touching event using a *single pre-trained pressure threshold*, and align it with the ground-truth touching event. The ground truth and ITrackU's time overlap (intersection of union) is reported as accuracy in Fig. 18. In most cases, pressure-based touch detection achieves over 90% accuracy, when there are no breaks or only a few breaks in the writing (1-S, 2-S, 3-S, or free-form). The dashed line tests our limits where the accuracy drops to 72%, most of which are false positives. We suspect this occurs since the fast and precise lifting and touching action in a dashed line leads the pressure exerted on the pen to not change enough for detection. We leave further improvements to future work.

Accuracy under different SNR: As with most wireless systems, ITrackU's accuracy will depend on the signal strength. We perform the tracking task across a wide range of signal strengths to obtain its working range We use a stencil when writing to make sure that exactly the same movement is compared under different SNRs. The power is adjusted by changing the transmitter's power gain. At the receiver, we collect CIR for every packet transmission and translate the amplitude of CIR into RSS [4]. Fig. 19 shows the median, the 5%*tile* and the 95%*tile* error under different RSS. ITrackU works stably with no visible impact for signal strength in excess of $-98dBm$. Errors increase when the RSS drops to $-101dBm$. By analyzing the CIR in low-SNR scenarios, we find that at least one of the three antenna's reception is too weak leading to frequent peak locating error. Further, long-term poor CIR cannot be compensated by IMU. ITrackU's tracking accuracy remains acceptable only over $-98dBm$ received signal strength.

6.2 Micro Benchmarks: UWB Tracking

In this part, we focus on performance evaluation when only UWB is used. We expose the temporal and spatial properties of UWB tracking, thereby showing the importance of incorporating IMU.

Temporal property: *Will ITrackU's accuracy degrade as time passes?* To evaluate ITrackU's performance over time, we repeatedly trace a stencilled rectangular pattern ($37cm \times 30cm$) on the touch screen for 5 minutes. Fig. 20 shows that over this period, the error is uniformly distributed with no visible drift. The actual trace is shown as an inset. As expected, ITrackU's accuracy does not degrade with time, indicating that ITrackU is potentially capable of long-term tracking in the real-world.

Spatial property: *Are there pockets of poor performance over the experimental area?* ITrackU's accuracy is influenced by many real-world problems faced by wireless signals, such as attenuation and multipath. Spatial interference is possible, which leads to location dependent performance. To answer this question, we split the experimental region into $10\,cm \times 10\,cm$ grids and measure the errors incurred in each grid. Fig. 21 shows a matrix of positional variation in accuracy. We note that accuracy does vary over the whole region. While most of grids have less than $10mm$ error, there are also high-error grids ($> 2cm$) distributed sporadically. CIR analysis reveals two causes for increased errors. First, in some corner areas, one of the antennas receives disproportionately high signal strength, which causes errors in CIR peak-finding. Second, at certain spots, the true CIR first-peak merges with a close multipath, polluting the phase readings. This positional variation highlights the insufficiency of a UWB-only approach. Combining UWB and IMU benefits from taking advantage of each other's strengths to improve accuracy. When the tracked object passes through high-error locations, the IMU dominates the tracking for a short duration, thereby mitigating the effect of such locations.

6.3 Micro Benchmarks: Fusion

In this part, we focus on the performance evaluation of the fusion algorithm under various scenarios.

Effect of writing speed: *How does writing speed affect performance?* On one hand, UWB tracking experiences wrapping error if the user writes too fast, where the IMU helps correct the wrong phase by estimating number of wraps from the motion data. On the other hand, due to the high-pass filter applied on IMU, IMU is more accurate when the user is writing fast. These two conflicting properties seemingly form a writing speed trade-off. To quantify performance at different speeds, we draw a rectangle on the touch screen at different speeds. We place RF absorbers in the field to remove the effect of potential multipath. The touch screen provides a stream of ground truth trace and the writing speed. Fig. 22 shows

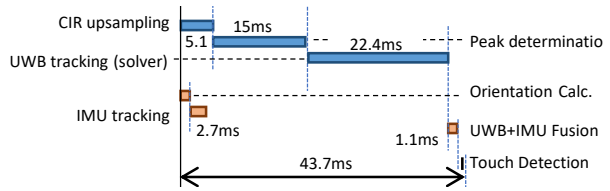

Figure 27: Time consumption of each function in ITrackU.

the 90^{th} percentile error when using ITrackU's fusion approach and contrasts it with the UWB-only approach. The x-axis denotes the maximum instantaneous speed during writing. Among these speeds, $0.6m/s - 2.1m/s$ are usual writing speeds. When the writing speed is $1m/s$ or smaller, the two approaches achieve comparable and low error, where the IMU does not give much benefit because no wrapping error occurs in this range. Wrapping occurs when we increase the writing speed to $2.1m/s$, which leads to a dramatic jump to $44mm$ 90^{th} percentile error with the UWB-only approach. This error further jumps to $138mm$ at $3m/s$ speed. Compared to the UWB-only approach, ITrackU achieves sub-centimeter level error ($4.4mm$ and $3.6mm$) after IMU-based compensation. This demonstrates ITrackU capability to sustain high accuracy even when a user writes extremely fast. Fig. 23 demonstrates the output when writing is very fast, with just UWB and with UWB-IMU fusion.

Tracking different sizes of characters: We evaluate the performance of ITrackU with varying sizes of tracking patterns, which matters in many applications. In the remote class example, the tracked characters or patterns can have a size ranging from $10cm$ to $1m$. We choose four sizes for evaluation: squares with edge length of $10cm$, $30cm$, $60cm$ and $1m$. Since these sizes exceed the touch screen size, we take a picture of the drawn pattern and then remove the projection effect and affine transformation caused by camera. This picture functions as the ground truth. The tracking results are then fit onto the picture, and the distance from every sample to the ground truth square is reported as error. Fig. 24 shows the error across different fontsizes. All the three sizes achieve a *mm*-level median error, which are $1.7mm$, $3.1mm$, $2.5mm$ and $2.7mm$, respectively. There is no significant increase in error as the fontsize increases to $1m$ showing ITrackU's ability to track a wide range of fontsizes without error accumulation.

Continuous working time of IMU: *How long can IMU compensate loss of UWB?* ITrackU allows IMU to take over tracking during low-confidence CIR (LC-CIR) durations. Given that IMU drifts, we must understand the limits of how long we can rely on the IMU when we keep suffering from poor UWB CIR quality. We use a trace-driven emulation to evaluate this. All the received CIRs and IMU data are stored offline. In the emulation, we artificially mark successive k UWB packets as LC-CIR forcing only IMU data to be used during this time. After the low confidence duration ends, we go back to track with UWB for a certain time, and then again enter the LC-CIR state. By varying the value of k, we obtain an emulated tracking error under different continuous *IMU-only* durations. Fig. 25 shows error increases with increasing lengths of LC-CIR durations. When the LC-CIR remains under $0.5s$, the overall error is less than $10mm$. When the duration lies in 0.5-1.5s, the error increases to $1cm$-$3cm$. While $3cm$ errors distort drawn pattern, ITrackU can still recover when high quality UWB packets are received, so long the APD error is within $\lambda/2 = 3.75cm$. Beyond that, irrecoverable tracking loss will result and UWB tracking must be reset using time-gaps between CIR peaks.

Effect of surfaces: *Does ITrackU perform equally well on different surfaces?* We evaluate ITrackU on different surfaces representing a variety of smoothness and textures: whiteboard, cardboard, wood, and blanket. Since the touch screen is unavailable for this, we take a picture as the ground truth as we do in the character-size micro-benchmark. To make sure that IMU takes effect, we manually induce LC-CIR durations by quickly sliding an aluminium shield across the direct signal path. Fig. 26 compares the 90^{th} percentile error of the UWB-only approach with ITrackU on all four surface-types. Results show that our fusion approach is a significant improvement over the UWB-only solution on the white board, card board, and the wooden board. The 90^{th} percentile error improves by about 4× on these surfaces. However, when writing on a blanket, ITrackU fails to correct the error during LC-CIR exposing the limits of ITrackU on a very rough surface. Deeper analysis shows that the constant high frequency wobbles on the rough blanket surface produces high frequency noise which makes IMU data erroneous. The IMU thus fails to aid in this case.

Time consumption: The overall ITrackU pipeline involves various components whose timing requirements and parallelism possibilities are now explored. We divide the processing pipeline into 7 functions: CIR upsampling, peak determination, UWB tracking, orientation calculation, IMU tracking, fusion, and touch detection. Data collection and transmission are not included since they are concurrent functions running much faster at a different end. Fig. 27 shows the time consumption in each function. Of all the functions, CIR peak determination and UWB tracking demand most time resources, which are $15ms$ and $22.4ms$ respectively. In contrast, the summation of time consumed by all other functions is less than $10ms$. The whole pipeline takes about $43.7ms$, yielding 22Hz sampling rate. We leave further pipeline optimizations to future work.

Signal blocking test: While signal blocking is an important problem in wireless signal propagation in general, blocking rarely happens when one writes holding the pen normally, given that the ITrackU transmitter is fixed on the pen's base (Fig. 28(a)). Yet a signal blocking study is still important to comprehensively understand the performance of ITrackU in harsh environments, and potentially in other use-cases for ITrackU. We tested two different holding styles which will block the signal shown in Fig. 28(b) and (c). We observe that complete blocking (c) degrades ITrackU into complete failure with no packets received at the receiver; there is nothing to study in that case. When partially blocking (like Fig. 28(b)), the signal is still detectable albeit with attenuated amplitude, which makes for an interesting case-study. To study the effects of partially blocking the signals, we split the whole experimental region into four sub-regions (labeled R1–R4 in Fig. 29). In each region, we perform 3 experiments—in each experiment, the transmitter's signal to a certain antenna (labeled AT1–AT3) is blocked by the palm of the hand. The transmitter power gain is set at 18.5dB[5]. The per-sub-region per-antenna tracking error is measured and compared in Fig. 30. When ITrackU fails to localize, we label it with "×". We observe that each sub-region has a "bottleneck" antenna blocking which causes tracking to fail (AT1 in regions R2, R3 and

[5]Refer to the Transmit Power Calibration and Management application note [15].

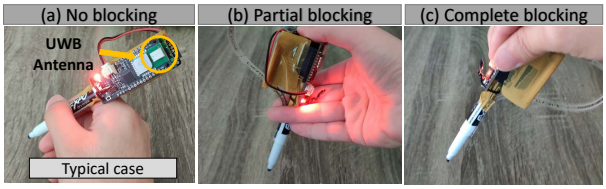

Figure 28: Blocking cases under different pen holding styles

Figure 31: ITrackU attached on top of Mimio.

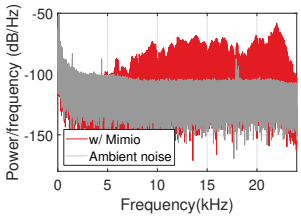

Figure 32: Mimio generates constant audible noise.

System	Technology	Error (50%)	Cost	Weight	Noise
ITrackU	UWB+IMU	2-3mm	$200	34g	No
Mimio	IR+Ultrasound	<1mm	$950	60g	Yes

Table 4: Comparison between Mimio and ITrackU.

Methodology	Unit size	Compression rate	Data size (1h)
ITrackU	10	None	1.03MB
480p video	848 × 480	200:1	213.6MB
720p video	1280 × 720	200:1	474.6MB

Table 5: Comparison between video streaming and ITrackU.

AT2 in regions R1, R4). This fundamentally results from the fact that blocking the signal with longer propagation distance attenuates it more significantly. However, an interesting observation is that blocking AT3 has almost no effect in any region; though it should have failed in R1. Recall that we attenuate the earlier antenna's signals, hence the signal corresponding to AT3 has the largest relative gain which counteracts the effect of blockage and attenuation in the air. This observation leads to a revelation: if in a particular use-case, a certain location is more likely to be blocked, it is beneficial to place AT3 in that location.

Charge cycle: *How often would we need to recharge ITrackU?* The receiver of ITrackU is connected to a continuous power source, hence we only consider the charge cycle at the transmitter (pen's attachment). We power the transmitter with LiPo battery LP503035 (3.7V, 500mAh) and allow it to run continuously (collecting sensor data and sending packets) until the battery is drained out. We observe a 10.8 hours life before the battery drains completely. This battery is 35 × 30 × 6mm; an acceptable size for a thick pen. Of course, we can choose smaller batteries to fit into slimmer pens, by trading off charging cycle. For instance, LP401230 (32×12×4.2mm) can run for 2.2-hours, and is small enough to fit into most pens.

Comparison with a commercial product: We compare our work with the commercial product MimioCapture Ink Recorder [14] (abbreviated as Mimio below). Mimio uses ultrasound and infrared signals jointly to track the movement of a pen. We affix the ITrackU's transmitter on the Mimio pen-sleeve (see Fig. 31) and draw different shapes on the whiteboard to test the performance of both systems. Points of major differences are listed in Table. 4. We find that Mimio is more accurate in tracking the pen's movements, but performs

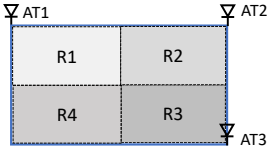

Figure 29: Four sub-regions in the partial blocking test.

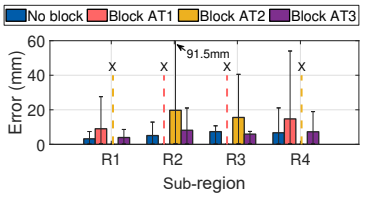

Figure 30: Tracking error of partial blocking (× indicates failure).

poorly in other aspects. ITrackU provides a lightweight and cheaper solution. Further, Mimio's ultrasound transmitter produces audible noise in the frequency range of 5K-20KHz—Fig. 32 shows the noise recorded by a regular microphone when Mimio is in use.

Storage and bandwidth efficiency: We use an online lecture example to show ITrackU's efficiency. Instead of video streaming a whiteboard's contents, ITrackU streams only the locations of the pen's tip. We use 10 bytes to encode a trace (4B for X/Y location, 4B for the time, 2B reserved for color/thickness). For one hour of writing, Table 5 compares the data size and streaming rate using ITrackU with different video qualities, with a 30Hz update rate. Low and standard resolution video requires about 213.6MB and 474.6MB data, respectively, over the whole lecture. In contrast, ITrackU consumes only 1.03MB to transmit the data, resulting in a much more efficient streaming process. The above computation projects ITrackU as a potential educational delivery mechanism even in communities with severely limited Internet access.

7 CONCLUDING REMARKS

ITrackU addresses fundamental problems in fine-grained wireless tracking, presenting a sensor-fusion approach in digitizing an instrument's movements on an unmodified surface. There are a number of potential extensions of this work. First, tracking can be enhanced by smoothing the output trajectory (such as using [40]) and by interpreting the written text or shapes drawn (such as in [36]). Second, the magnetometer readings can be corrupted by external magnetic fields which need to be compensated for through calibration. The use of the IMU is central to orientation and tilt tracking; its effectiveness is influenced by the quality of the sensor data. The problem of resolving the effect of dynamic magnetic field variation using other approaches should, therefore, be addressed in future work. Third, wireless signals can inherently leak information. While UWB packets can be encrypted (IMU and pressure sensor information), the CIR, which is derived from the preamble process, cannot be encrypted. By installing a multi-antenna UWB receiver nearby, listening on the same channel, one can potentially snoop on the pen's movements from outside a closed room. Protecting CIR data is an open research problem, which we leave to future work.

In summary, ITrackU presents an approach of combining UWB, as the fundamental tracking technique, with IMU, to bring multi-facet benefits in accuracy, range, robustness, and cost, to the problem of tracking a pen-like instrument.

Acknowledgments

We thank the anonymous reviewers and our shepherd for providing valuable feedback. This work has been partially supported by the NSF under Grant No. 2031868.

REFERENCES

[1] 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.

[2] Magnetic calibration algorithms, 2016. https://www.nxp.com/docs/en/application-note/AN5019.pdf.

[3] 4-wire analog resistive touch screen datasheet, 2017. https://media.digikey.com/pdf/Data%20Sheets/NKK%20PDFs/FT_Series_4-Wire_Ds_Oct_2017.pdf.

[4] Decawave user manual, 2017. https://www.decawave.com/sites/default/files/resources/dw1000_user_manual_2.11.pdf.

[5] Microsoft surface pen, 2017. https://news.microsoft.com/uploads/2017/05/SurfacePenFS.pdf.

[6] Apple pencil, 2018. https://www.apple.com/apple-pencil/.

[7] S pen, 2018. https://www.samsung.com/global/galaxy/galaxy-note10/s-pen/.

[8] MAVL: Multiresolution analysis of voice localization. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, Boston, MA, April 2021. USENIX Association.

[9] Fadel Adib, Zach Kabelac, Dina Katabi, and Robert C Miller. 3d tracking via body radio reflections. In *11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14)*, pages 317–329, 2014.

[10] Fadel Adib and Dina Katabi. See through walls with wifi! In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pages 75–86, 2013.

[11] Yang Bai, Li Lu, Jerry Cheng, Jian Liu, Yingying Chen, and Jiadi Yu. Acoustic-based sensing and applications: A survey. *Computer Networks*, 181:107447, 2020.

[12] 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.

[13] Yifeng Cao, Ashutosh Dhekne, and Mostafa Ammar. 6fit-a-part: A protocol for physical distancing on a custom wearable device. In *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, pages 1–12. IEEE, 2020.

[14] Boxlight Corporation. Mimiocapture ink recorder datasheet, 2016. https://mimio.boxlight.com/wp-content/uploads/2017/03/MimioCapture_SellSheet.pdf.

[15] Decawave. Transmit power calibration and management, 2019. https://www.decawave.com/wp-content/uploads/2019/07/APS023_Part-1_Transmit_Power_Calibration__Management.pdf.

[16] 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.

[17] Tobias Feigl, Sebastian Kram, Philipp Woller, Ramiz H Siddiqui, Michael Philippsen, and Christopher Mutschler. Rnn-aided human velocity estimation from a single imu. *Sensors*, 20(13):3656, 2020.

[18] 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.

[19] 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.

[20] Bernhard Großwindhager, Michael Rath, Josef Kulmer, Mustafa S Bakr, Carlo Alberto Boano, Klaus Witrisal, and Kay Römer. Salma: Uwb-based single-anchor localization system using multipath assistance. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 132–144, 2018.

[21] Pengfei Guo, Haitao Qiu, Yunchun Yang, and Zhang Ren. The soft iron and hard iron calibration method using extended kalman filter for attitude and heading reference system. In *2008 IEEE/ION Position, Location and Navigation Symposium*, pages 1167–1174. IEEE, 2008.

[22] 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.

[23] Evan G Hemingway and Oliver M O'Reilly. Perspectives on euler angle singularities, gimbal lock, and the orthogonality of applied forces and applied moments. *Multibody System Dynamics*, 44(1):31–56, 2018.

[24] KC Ho and Wenwei Xu. An accurate algebraic solution for moving source location using tdoa and fdoa measurements. *IEEE Transactions on Signal Processing*, 52(9):2453–2463, 2004.

[25] Ngoc-Huynh Ho, Phuc Huu Truong, and Gu-Min Jeong. Step-detection and adaptive step-length estimation for pedestrian dead-reckoning at various walking speeds using a smartphone. *Sensors*, 16(9):1423, 2016.

[26] Fang Hongzhao. *60 GHz RSS localization with omni-directional and horn antennas*. PhD thesis, 2010.

[27] Carl Q Howard, Colin H Hansen, and Anthony C Zander. A review of current ultrasound exposure limits. *The Journal of Occupational Health and Safety of Australia and New Zealand*, 21(3):253–257, 2005.

[28] Wenchao Huang, Yan Xiong, Xiang-Yang Li, Hao Lin, Xufei Mao, Panlong Yang, and Yunhao Liu. Shake and walk: Acoustic direction finding and fine-grained indoor localization using smartphones. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 370–378. IEEE, 2014.

[29] Chengkun Jiang, Junchen Guo, Yuan He, Meng Jin, Shuai Li, and Yunhao Liu. mmvib: micrometer-level vibration measurement with mmwave radar. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–13, 2020.

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

[31] Wei Li and Jinling Wang. Effective adaptive kalman filter for mems-imu/magnetometers integrated attitude and heading reference systems. *The Journal of Navigation*, 66(1):99–113, 2013.

[32] Hongbo Liu, Yu Gan, Jie Yang, Simon Sidhom, Yan Wang, Yingying Chen, and Fan Ye. Push the limit of wifi based localization for smartphones. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 305–316, 2012.

[33] Yang Liu, Zhenjiang Li, Zhidan Liu, and Kaishun Wu. Real-time arm skeleton tracking and gesture inference tolerant to missing wearable sensors. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 287–299, 2019.

[34] Robert Mahony, Tarek Hamel, and Jean-Michel Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on automatic control*, 53(5):1203–1218, 2008.

[35] 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.

[36] Microsoft. Change handwritten ink to shapes, text, or math in powerpoint for microsoft 365, 2020. https://support.microsoft.com/en-us/topic/change-handwritten-ink-to-shapes-text-or-math-in-powerpoint-for-microsoft-365-0740dec3-6291-4c1f-8baa-011d18449919.

[37] 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.

[38] Abhinav Parate, Meng-Chieh Chiu, Chaniel Chadowitz, Deepak Ganesan, and Evangelos Kalogerakis. Risq: Recognizing smoking gestures with inertial sensors on a wristband. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 149–161, 2014.

[39] Neal Patwari, Lara Brewer, Quinn Tate, Ossi Kaltiokallio, and Maurizio Bocca. Breathfinding: A wireless network that monitors and locates breathing in a home. *IEEE Journal of Selected Topics in Signal Processing*, 8(1):30–42, 2013.

[40] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer graphics and image processing*, 1(3):244–256, 1972.

[41] 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.

[42] Nirupam Roy, Sheng Shen, Haitham Hassanieh, and Romit Roy Choudhury. Inaudible voice commands: The long-range attack and defense. In *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)*, pages 547–560, 2018.

[43] 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.

[44] Sheng Shen, He Wang, and Romit Roy Choudhury. I am a smartwatch and i can track my user's arm. In *Proceedings of the 14th annual international conference on Mobile systems, applications, and services*, pages 85–96, 2016.

[45] Ke Sun, Wei Wang, Alex X Liu, and Haipeng Dai. Depth aware finger tapping on virtual displays. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pages 283–295, 2018.

[46] Ke Sun, Ting Zhao, Wei Wang, and Lei Xie. Vskin: Sensing touch gestures on surfaces of mobile devices using acoustic signals. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 591–605, 2018.

[47] Emily Tate. Sensonor stim3000, June 2020. https://www.sensonor.com/products/inertial-measurement-units/stim300/.

[48] David Tse and Pramod Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

[49] He Wang, Ted Tsung-Te Lai, and Romit Roy Choudhury. Mole: Motion leaks through smartwatch sensors. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 155–166, 2015.

[50] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. No need to war-drive: Unsupervised indoor localization. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 197–210, 2012.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] Chao Xu, Parth H Pathak, and Prasant Mohapatra. Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, pages 9–14, 2015.

[57] Chenren Xu, Bernhard Firner, Robert S Moore, Yanyong Zhang, Wade Trappe, Richard Howard, Feixiong Zhang, and Ning An. Scpl: Indoor device-free multi-subject counting and localization using radio signal strength. In *Proceedings of the 12th international conference on Information processing in sensor networks*, pages 79–90, 2013.

[58] 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.

[59] Seanglidet Yean, Bu Sung Lee, Chai Kiat Yeo, and Chan Hua Vun. Algorithm for 3d orientation estimation based on kalman filter and gradient descent. In *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 1–6. IEEE, 2016.

[60] Huanpu Yin, Anfu Zhou, Guangyuan Su, Bo Chen, Liang Liu, and Huadong Ma. Learning to recognize handwriting input with acoustic features. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(2):1–26, 2020.

[61] Tuo Yu, Haiming Jin, and Klara Nahrstedt. Mobile devices based eavesdropping of handwriting. *IEEE Transactions on Mobile Computing*, 2019.

[62] 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.

[63] 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.

[64] Bing Zhou, Jay Lohokare, Ruipeng Gao, and Fan Ye. Echoprint: Two-factor authentication using acoustics and vision on smartphones. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 321–336, 2018.