Strata: Fine-Grained Device-Free Tracking Using Acoustic Signals

ABSTRACT

Next generation devices, such as virtual reality (VR), augmented reality (AR), and smart appliances, demand a simple and intuitive way for users to interact with them. To address such needs, we develop a novel acoustic based device-free tracking system, called Strata, to enable a user to interact with a nearby device by simply moving his finger. In Strata, a mobile (*e.g.*, smartphone) transmits known audio signals at an inaudible frequency, and analyzes the received signal reflected by the moving finger to track the finger location. To explicitly take into account multipath propagation, the mobile estimates the channel impulse response (CIR), which characterizes signal traversal paths with different delays. Each channel tap corresponds to the multipath effects within a certain delay range. The mobile selects the channel tap corresponding to the finger movement and extracts the phase change of the selected tap to accurately estimate the distance change of a finger. Moreover, it estimates the absolute distance of the finger based on the change in CIR using a novel optimization framework. We then combine the absolute and relative distance estimates to accurately track the moving target. We implement our tracking system on Samsung S4. Through micro-benchmarks and user studies, we show that our system achieves high tracking accuracy and low latency without extra hardware.

1. INTRODUCTION

Motivation: Smart appliances, Virtual Reality (VR), and Augmented Reality (AR) are all taking off. The availability of easy-to-use user interface is the key to their success. Smart TVs are still cumbersome to navigate through the menus. Many smart appliances require users to manually launch smartphone applications and click through, which is even more cumbersome than actually turning on/off switches. VR and AR are expected to hit $150 billion by 2020. They provide immersive experience, and open the doors to new ways of training, education, meeting, advertising, travel, health care, emergency responses, and scientific experiments. However, the current user interfaces of VR/AR are rather limited: they rely on tapping, swiping, voice recognition, or steering the camera towards the hand to make sure the hand is within the view and line-of-sight of the camera while wearing the headset. Our vision is to develop a device-free user interface (UI) so that a user can freely move his or her hand to control game consoles, VR, AR, and smart appliances.

Directly tracking hand or finger movement without any device is appealing due to convenience. We have interviewed VR, AR, game developers and users. They all prefer device-free based user interface (UI) (*i.e.*, controlling devices directly using hands without holding anything) since it is cumbersome to control a device outside the view in VR/AR and it is more natural to play games and interact with VR/AR objects using hands directly.

Challenges: Enabling accurate device-free tracking is particularly challenging. In such a case, reflected signal has to be used for tracking. Reflected signal is much weaker than the directly received signal (*e.g.*, in free space, the directly received signal attenuates by $1/d^2$ whereas the reflected signal attenuates by $1/d^4$, where d is the distance between the device and the target to be tracked). Moreover, it is more difficult to handle multiple reflection paths in device-free tracking. In device-based tracking, one may rely on finding the first arriving signal since the straight-line path between the sender and receiver is shortest. In comparison, in device-free tracking, the path(s) of interest is the shortest, which makes it even harder to distinguish which path should be used for tracking.

Recently, there have been considerable work on device-free motion tracking using various signals, such as wireless signals (*e.g.*, [12, 3, 40, 31]), acoustic signals (*e.g.*, [21, 29]), and vision [1, 2, 36, 33]). However, they either require specialized hardware or provide insufficient accuracy. For example, WiDeo [12] that is a recent research on WiFi-based tracking pushes the tracking accuracy to up to a few centimeters, but it requires full-duplex wireless hardware. WiTrack [3] tracks user position with 10-13 cm error, but it uses customized hardware that transmits chirp signal through 1.67 GHz bandwidth. WiDraw [31] achieves 5 cm tracking accuracy with the support of 25 WiFi APs around the user. Therefore, they require complicated hardware while providing insufficient accuracy to enable motion-based UI for VR/AR users. 60GHz-based tracking (*e.g.*, [40]) achieves higher accuracy but requires 60 GHz antenna arrays, which are not widely available.

Recently, two state-of-the-art acoustic tracking schemes (*i.e.*, FingerIO [21] and LLAP [29]) enable device-free tracking only using the existing speaker and microphones in mobile device. Although they achieve higher accuracy than the other schemes, due to their vulnerability on the multi-path reflections and surrounding movements, their tracking accuracy is still limited. For example, based on our extensive experiment, LLAP [29] achieves 0.7 cm distance estimation error in 1D tracking, but its trajectory error in 2D space increases to 2.1 cm. When there are moving people around the tracking object, the accuracy gets worse. The accuracy of FingerIO is even lower than LLAP.

Therefore, despite significant progress, achieving highly accurate and responsive device-free tracking on commodity hardware remains an open challenge. According to personal communication with game and application developers, sub-centimeter level accuracy and within 16 ms response time are required in order to provide good user experience. This is especially challenging to achieve using a commodity device, such as a smartphone, given its limited processing power and lack of special hardware.

Our approach: Built on the existing work, we develop a new device-free tracking using acoustic signals. In our system, a mobile (*e.g.*, smartphone) continuously transmits inaudible acoustic signals. The signals are reflected by nearby objects, including a moving finger, and arrive at the microphone on the same mobile. The mobile analyzes the received signal to estimate the channel, based on which it estimates the distance change and absolute distance to locate the finger.

Due to the small wave-length of acoustic signals, it is promising to derive the distance change based on the phase. Phase is also more robust to imperfect frequency response of a speaker. However, like many wireless signals, audio signals go through multiple paths to reach the receiver (*e.g.*, due to reflection by different objects). Such multipath propagation poses significant challenges to using the phase of the raw received signal for tracking. To address the challenge, we estimate channel impulse response (CIR) in the time-domain. The estimate gives the channel coefficient of each channel tap. We then select an appropriate channel tap and use the phase of the selected tap to estimate the distance change of a finger.

To further derive the absolute distance, we develop a novel framework to estimate the absolute distance of the path reflected by the moving finger during a few consecutive intervals such that its changes match with the changes in the CIR during these intervals and the distance changes between the intervals match with the phase measurement. Inferring the absolute distance serves two purposes: (i) it allows us to get the initial absolute distance so that we can translate the subsequent distance change into a new absolute distance, and (ii) it can be used to improve the tracking accuracy and alleviate error accumulation in subsequent intervals by combining it with the relative distance change.

We implement our approach on Samsung S4, which has one speaker and two microphones, and enable real-time tracking of the user's moving finger. With the extensive user-study, we show our system has three distinct features: (i) high accuracy: within 0.3 cm distance tracking error, 1.0 cm 2D tracking error, and 0.6 cm drawing error in a 2D space; (ii) low latency: we can update the position every 12.5ms, and (iii) easy to deploy: a smartphone can track a nearby finger movement without extra hardware.

2. OUR APPROACH

In this section, we present a fine-grained acoustic-based device-free tracking.

2.1 Overview

We use the phase change of the acoustic channel to estimate the distance change. This allows us to achieve high accuracy because the acoustic wavelength is very short. For example, the wavelength is 1.9 cm in 18 KHz audio frequency. Only 1 mm movement causes the reflected path length to change by 2 mm, which results in 0.21π phase change, large enough to detect.

However, in practice, due to multi-path propagation (*i.e.*, a signal traverses multiple paths before arriving at the receiver), the impact of a moving target on the overall channel can be very complicated, and varies across environments. In order to address this challenge, we use the phase from the estimated channel impulse response (CIR) rather than the raw received signal. CIR is a characterization of all signal traversal paths with different delays and magnitudes [27]. Specifically, it is a vector of channel taps where each channel tap corresponds to multi-path effects within a specific delay range. By focusing on the phase change of certain channel taps whose delays are close to the target delay range, we can effectively filter out the phase change incurred by the movement of objects outside a certain range as determined by the number of taps being used.

The phase change only gives us the distance change. We need to know the absolute distance at some point in order to translate the distance change into an absolute distance for tracking. Moreover, using the distance change alone incurs error accumulation, since the distance at a given time is estimated as the sum of all previous distance changes plus the initial position, each of which may incur an error. To address both issues, we develop a technique to estimate the absolute distance, which is used to get the initial position and also enhance the tracking accuracy by combining it with the distance change over time.

Putting together, our overall system consists of the following steps, which we will elaborate in this section.

1. Estimate channel impulse response (CIR) (Section 2.3);
2. Identify the channel tap corresponding to the target, and track the phase change of the selected tap in CIR to estimate the distance change (Section 2.4);
3. Estimate the absolute distance based on CIR (Section 2.5);
4. Combine the absolute distance with the relative distance to get a more accurate distance estimate, and track the target's position based on the distance to different landmarks (*e.g.*, microphones) (Section 2.6).

Our system implementation uses 18-22 KHz frequency and 48 KHz sampling rate. We can easily support other bandwidths and sampling rates.

2.2 Background

A wireless signal, including an audio signal, travels through a straight line from the transmitter to the receiver in free space. In reality, due to obstacles in the environment, a single transmitted signal will reach the receiver via multiple paths (*e.g.*, paths going through different reflectors). Therefore, the received signal is a superposition of multiple signals with different delays. The received signal via multipath is traditionally modeled as the following Linear Time-Invariant

(LTI) system [34]. Suppose the channel has L paths and the received signal from path i has delay τ_i and amplitude a_i determined by the travel distance of the path and reflectors. Then, the received signal $y(t)$ is the summation of L signals, as shown below:

$$y(t) = \sum_{i=1}^{L} a_i x(t-\tau_i) = \sum_{i=1}^{L} a_i e^{-j2\pi f_c \tau_i} s(t-\tau_i) = h(t) * x(t),$$

where $s(t)$ and $x(t)$ are the transmitted baseband and the passband signals at time t, respectively, and $h(t)$ is the channel impulse response. $h(t) = \sum_{i=1}^{L} a_i e^{-j2\pi f_c \tau_i}\delta(t-\tau_i)$, where $\delta(t)$ is Dirac's delta function [9].

The channel estimate from the received baseband symbols is a discrete output of $h(t)$ sampled every T_s [34], which is

$$h[n] = \sum_{i=1}^{L} a_i e^{-j2\pi f_c \tau_i}\,\mathrm{sinc}(n-\tau_i W), \tag{1}$$

where $\mathrm{sinc}(t) = \frac{\sin(\pi t)}{\pi t}$. Conventionally, $h[n]$ is called the n-th channel tap, because CIR is regarded as a discrete-time filter in LTI system. Note that sinc function decays over time, so the impact of delayed signal on the measured $h[n]$ is small when the difference between nT_s and τ_i are sufficiently large. However, if they are relatively close, movement is captured in multiple channel taps due to the signal dispersion effect of *sinc* function. This is illustrated in Figure 1, where there is only one small object within 50 cm away from the mobile and the object is 30cm away closest to tap 3 ($h[3]$), but the nearby channel taps also see non-negligible magnitude.

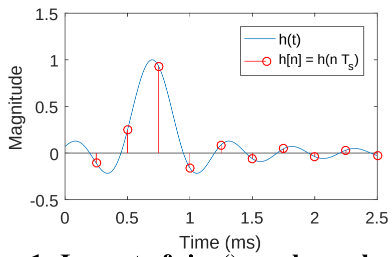
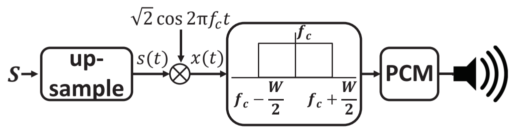


Figure 1: Impact of sinc() on channel estimate.

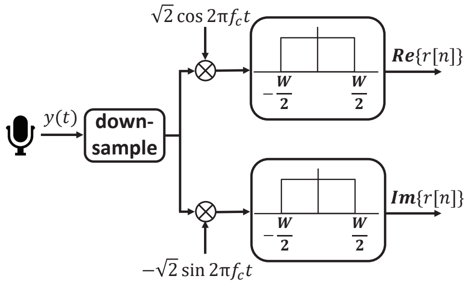
2.3 Estimating Channel Impulse Response

Single-carrier communication channel: To estimate the channel, we design data communication for acoustic channel. An important design decision is whether we should use single-carrier or multi-carrier (*e.g.*, OFDM) communication. OFDM is widely used in modern wireless communication due to its efficiency and robustness to Inter Symbol Interference (ISI) caused by multipath. However, it yields channel estimation in frequency domain, while channel estimate is often more useful in time domain for tracking and localization purpose [19]. So we need to transform channel coefficients from frequency domain to time domain [28], but this process requires additional computation and may incur significant noise [41]. Therefore, we use a single-carrier based communication system to estimate the channel in time domain without extra processing.

Transmission signal design: A transmitter sends a known training sequence for channel estimation. Let $\mathbf{S} = \{s[1], ..., s[K]\}$ denote the training sequence, where K is the length of the sequence. It can be any random bits. We choose 26-bit GSM training sequence because it is known to have good properties for synchronization and channel estimation [25] and widely used in single carrier communication. We modulate $\mathbf{S}$ to BPSK symbols, where bits 0 and 1 are mapped to baseband symbols 1 and -1, respectively.



(a) Transmitter design



(b) Receiver design

Figure 2: Transmitter and Receiver system diagram.

Figure 2(a) illustrates signal generation and transmission process. To transmit a modulated symbol over the inaudible frequency band, we first need to reduce the signal bandwidth so that it does not exceed the maximum allowed bandwidth of the inaudible band. Let f_s and B denote the sampling rate and the channel bandwidth, respectively. To limit the bandwidth of the transmitted symbol, we upsample the symbol with a rate of $\frac{f_s}{B}$, which is done by zero padding and low-pass filtering to smooth discontinuity [22]. Finally, we up-convert the signal to transmit it over the inaudible band. Let f_c be the center frequency of the passband. We change the frequency of the signal by multiplying $\sqrt{2}\cos(2\pi f_c t)$ to the baseband signal: $x(t) = \sqrt{2}\cos(2\pi f_c t)s(t)$, where $s(t)$ and $x(t)$ are upsampled baseband and passband signals, respectively. Since BPSK only has real parts, $x(t) = \sqrt{2}e^{-j2\pi f_c t}s(t)$.

To remove noise outside the transmission band, we perform band-pass filtering on $x(t)$ with pass-band range from $f_c - \frac{B}{2}$ Hz to $f_c + \frac{B}{2}$ Hz. The generated passband signals are transmitted through the smartphone speaker. Since the transmitted training sequence is always fixed, it can be generated offline and saved as a format of 16-bit Pulse Coded Modulation (PCM) in a Waveform Audio (WAV) file, which can be played by any mobile (*e.g.*, smartphone or smart watch).

We refer to the training sequence as a frame. Between frames, we insert a fixed gap (*i.e.*, zero symbols) to avoid inter-frame interference. The gap should be sufficiently long so that the delayed signal from the previous frame does not interfere with the new frame. However, it should be as short as possible to provide low latency. Our study shows 24 zero symbols between frames are sufficient. As a result, a frame has 50 symbols. Given that the baseband symbol interval is $\frac{1}{B} = 0.25$ ms, each frame lasts 12.5 ms. So we update new channel estimate and the target's position every 12.5ms, which is below 16 ms required for providing seamless user experience.

Receiver design: Figure 2(b) illustrates the signal reception and baseband conversion process. The received passband signal $y(t)$ arriving at the microphone is converted into a baseband symbol $r[n]$ using the following down-conversion process: $y(t)$ is multiplied by $\sqrt{2}\cos(2\pi f_c t)$ and $-\sqrt{2}\sin(2\pi f_c t)$ to get the real and imaginary parts of the received baseband symbol, respectively. We perform low-path filtering and down-sampling to select a signal every symbol interval. This gives us the following baseband signal [1]: $r[n] = \sqrt{2}\cos(2\pi f_c t)y(t) - j\sqrt{2}\sin(2\pi f_c t)y(t) = \sqrt{2}e^{-j2\pi f_c t}y(t)$, where t is the time that the n-th baseband symbol is sampled (*i.e.*, $t = n \times T_s$, where T_s is a symbol interval).

Frame detection: After passband-to-baseband signal conversion, the receiver detects the first symbol of the received frame by energy detection and cross-correlation. We first detect the rough beginning of the frame based on energy detection: if the magnitude of the 3 consecutive symbols are higher than the threshold σ, we treat the first symbol as the beginning of the frame symbols. Our implementation uses $\sigma = 0.003$. Then we find more precise starting point based on cross-correlation. Specifically, we find the sample that gives the maximum cross-correlation magnitude between the received and transmitted frames. Note the frame detection procedure is only necessary at the beginning of tracking. Since the frame interval is fixed, once a frame is detected, the subsequent frames can be determined by adding a constant frame interval.

Channel estimation: Next we estimate the channel based on the received frame and the known training sequence. There are several existing channel estimation algorithms in single carrier communication system. We use least-Squares (LS) channel estimation since it is cheap to compute on a mobile. We mainly focus on the algorithm implementation, and refer readers to [25] for the fundamental theory behind it.

For LS channel estimation, one needs to decide the reference length P and the memory length L, where L determines the number of channel taps we can estimate and $P+L$ is the training sequence length. Increasing L allows us to estimate more channel taps but reduces the reliability of estimation. Our implementation uses $P = 16$ and $L = 10$, which implies we can track movement up to 50 cm away (see Section 2.4.1). One can easily adapt P according to the environment.

Let $\mathbf{m} = \{m_1, m_2, \ldots, m_{L+P}\}$ denote the training sequence. A circulant training matrix $\mathbf{M} \in \mathbb{R}^{P \times L}$ is:

$$\mathbf{M} = \begin{bmatrix} m_L & m_{L-1} & m_{L-2} & \ldots & m_1 \\ m_{L+1} & m_L & m_{L-1} & \ldots & m_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{L+P} & m_{L+P-1} & m_{L+P-2} & \ldots & m_{P+1} \end{bmatrix}.$$

Let $\mathbf{y} = \{y_1, y_2, \ldots, y_{L+P}\}$ denote the received training sequence. The channel is estimated as

$$\hat{\mathbf{h}} = (\mathbf{M}^H\mathbf{M})^{-1}\mathbf{M}^H\mathbf{y_L}, \tag{2}$$

where $\mathbf{y_L} = \{y_{L+1}, y_{L+2}, \ldots, y_{L+P}\}$.

Given the pre-computed $(\mathbf{M}^H\mathbf{M})^{-1}\mathbf{M}^H$, the computational cost of the channel estimation is only the matrix-to-vector multiplication, which is $O(P \times L)$. Given $P = 16$ and $L = 10$, the channel estimation complexity is low enough to implement on a mobile.

Improving channel estimation accuracy: We further improve the channel estimation accuracy. In traditional digital communication, downsampling during the passband to baseband conversion simply picks one out of every r samples from the over-sampled signals, where r is the upsampling rate. In Strata, r is 12. So 11 samples out of 12 samples are thrown away in the downsampling process, which can be wasteful. Such wastage arises in our setting because the audio sampling rate is 48 KHz, which is much higher than the Nyquist rate 8KHz. Instead of simply dropping over-sampled signals, we average them to reduce noise in the output. From every r samples in a sampling interval, we pick the first l samples and use their average as the downsampled output. Our evaluation uses $l = 4$. We also evaluate the impact of l in Section 3.

2.4 Tracking Phase Change

2.4.1 Overview

Next we track the phase change based on CIR estimates. We study the impact of the reflected signal using the following experiment. We move a small aluminium ball ($<$ 1 cm diameter) attached to a long and thin wood stick. The person moving the stick is over 1 m away from the ball (outside the range of CIR taps). The ball is initially 30 cm away from the smartphone and moved 20 cm towards the phone. Figure 3 shows the phase of multiple channel taps while the ball is moving towards the smartphone. The result shows that the phase rotates in multiple taps, which indicates the path length is changing. This change is caused by the moving ball. As shown in Figure 3, even though only a single

[1]Conventionally, $(\cdot)$ and $[\cdot]$ notations are used to represent analog and digital signals, respectively. Here every signal is digital because a mobile app cannot access the analog signal. We use $(\cdot)$ and $[\cdot]$ notations to distinguish upsampled signal with rate f_s from the downsampled signal with rate B.

small object moves, the phase rotation is observed in multiple taps. We repeat the experiments multiple times, and find that the phase rotation is observed approximately in 3 consecutive taps and the reflected signal with delay τ affects the three $h[n]$ that have the smallest $|\tau - nT_s|$. As a result, $h[n]$ can be approximated as:

$$h[n] \approx \sum_k a_k e^{-j2\pi f_c \tau_k}, \qquad (n - \frac{3}{2})T_s < \tau_k < (n + \frac{3}{2})T_s. \tag{3}$$

In other words, each channel tap $h[n]$ contains the phase and magnitude of the reflected signals whose delays are between $(n - \frac{3}{2})T_s$ and $(n + \frac{3}{2})T_s$. The path length changes with the delay τ_k according to $d_k = \tau_k V_c$, where d_k is the travel distance of the path k and V_c is the propagation speed of the audio (*i.e.*, $V_c \approx 340m/s$). Assuming that the speaker and microphone are closely located, the distance from the microphone to the reflecting object is approximately half of the travel distance. Therefore, $h[n]$ indicates the object's distance from microphone is between $(n - \frac{3}{2})\frac{T_s V_c}{2}$ and $(n + \frac{3}{2})\frac{T_s V_c}{2}$. Given $T_s = 0.25$ ms, $\frac{T_s V_c}{2} = 4.25$ cm and each tap captures objects across 12.75 cm range. This enables us to filter out the movement of objects outside the target range. For example, if we want to track the movement of a finger within 50 cm from the mobile, we can limit the channel taps to the first 10 taps to filter out the movement outside 50 cm. This is because the 10th tap may contain information from objects up to around 12th taps away, which gives $12 * 4.25 = 51$ cm.

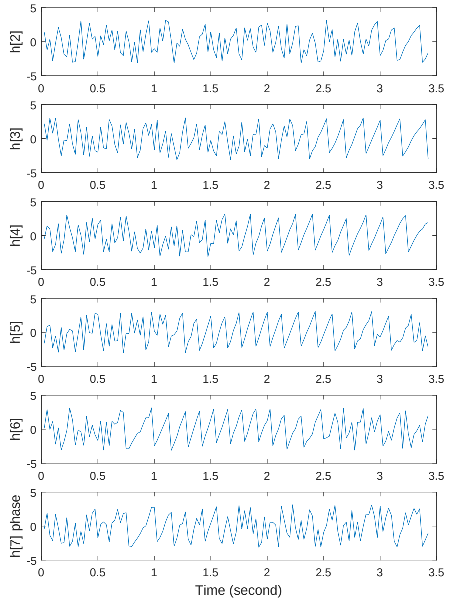


Figure 3: Phase change in multiple channel taps while moving a ball.

Next we track the phase change using the CIR estimate. While the CIR vector captures the channels with different propagation distances, it is challenging to extract the phase change caused by the target movement based on CIR since multiple paths with similar distances are mixed within each channel tap. To address the issue, we decompose the problem into the following two steps: (i) if we know which channel tap is affected by the moving target, how to extract the phase change caused by the target's movement, and (ii) how to determine which channel tap is affected by the target. Below we present our approaches to address both issues.

2.4.2 *Estimate Phase Change*

We assume the k-th channel tap is affected by the target's movement. In order to observe the phase change of the moving target, we compare the two consecutive channel measurements. Taking difference between the two consecutive channels effectively removes dominant static reflections. Let L_k denote the number of paths observed in $h[k]$. Suppose the L_k-th path is the path reflected from the moving finger, while the other $L_k - 1$ paths remain the same during the two consecutive channel measurement periods $t-1$ and t. Then,

$$h[k]^{t-1} = \sum_{i=1}^{L_k} a_i e^{-j2\pi f_c \tau_i(t-1)},$$

$$h[k]^{t} = \sum_{i=1}^{L_k-1} a_i e^{-j2\pi f_c \tau_i(t)} + a_{L_k} e^{-j2\pi f_c(\tau_{L_k}(t-1)+\tau_d(t))},$$

where $h[k]^t$ is the k-th channel tap estimated from the t-th frame and $\tau_d(t)$ is the delay difference caused by the target movement between the t-th and $(t-1)$-th frame intervals (*i.e.*, $\tau_d(t) = \tau_{L_k}(t) - \tau_{L_k}(t-1)$). By taking their difference, we get

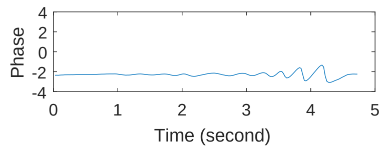
$$h_d[k]^t = a_{L_k}(e^{-j2\pi f_c(\tau_{L_k}(t-1)+\tau_d(t))} - e^{-j2\pi f_c \tau_{L_k}(t-1)}), \tag{4}$$

where $h_d[k]^t = h[k]^t - h[k]^{t-1}$. Equation 4 assumes that a_{L_k} associated with a propagation path is constant over two consecutive measurements due to a very small distance change in a 12.5 ms interval. From the angle of $h_d[k]^t$, we observe the phase rotation caused by the change of $\tau_{L_k}(t)$.

$$\begin{aligned}\angle(h_d[k]^t) &= \angle(e^{-j2\pi f_c(\tau_{L_k}(t-1)+\tau_d(t))} - e^{-j2\pi f_c \tau_{L_k}(t-1)}) \\ &= \angle(e^{-j2\pi f_c \tau_{L_k}(t-1)}(e^{-j2\pi f_c \tau_d(t)} - 1)) \\ &= \angle(e^{-j2\pi f_c \tau_{L_k}(t-1)}) + \frac{\angle(e^{-j2\pi f_c \tau_d(t)})}{2} + \frac{\pi}{2},\end{aligned} \tag{5}$$

where $\angle(X)$ is the phase of the complex number X. We can prove $\angle(e^{-j2\pi a} - 1) = \frac{\angle(e^{-j2\pi a})}{2} + \frac{\pi}{2}$ geometrically. The proof is omitted in the interest of brevity.

Figure 4 (a) and (b) show the phases of $h[k]$ and $h_d[k]$, respectively, while a user is moving his finger towards the speaker and microphone. In the collected trace, we conjecture $h[k]$ includes the finger movement related path between 1.0 second and 4.6 second. In Figure 4(a), the phase is very stable and the change by the finger movement is not clear because the majority portion of $h[k]$ contains signals from the


(a) Phase of $h[k]$

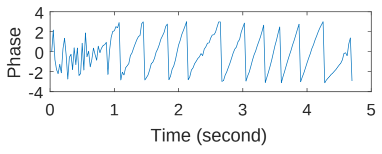

(b) Phase of $h_d[k]$

Figure 4: Phase of the channel impulse responses while a finger is moving.

static paths. After removing the impact of the static paths, we can observe clear phase rotation due to the finger movement from $h_d[k]$.

From the phase difference between $h_d[k]^{t+1}$ and $h_d[k]^t$, we get the phase rotation caused by the delay difference $\angle(e^{-j2\pi f_c \tau_d(t)})$, and eventually the travel distance of the finger during the measurement interval using the relation between the phase change. Note that $\tau_d(t) = \tau_{L_k}(t) - \tau_{L_k}(t-1)$. Using Equation 5, we represent the phase difference as

$$\angle(h_d[k]^{t+1}) - \angle(h_d[k]^t) = \angle(e^{-j2\pi f_c \tau_{L_k}(t)})$$
$$-\angle(e^{-j2\pi f_c \tau_{L_k}(t-1)}) + \frac{1}{2}(\angle(e^{-j2\pi f_c \tau_d(t+1)}) - \angle(e^{-j2\pi f_c \tau_d(t)}))$$
$$= \angle(e^{-j2\pi f_c \tau_d(t)}) + \frac{1}{2}(\angle(e^{-j2\pi f_c \tau_d(t+1)}) - \angle(e^{-j2\pi f_c \tau_d(t)})).$$

By solving the above equation, we can calculate $\angle(e^{-j2\pi f_c \tau_d(t)})$. Without prior, we can simply assume $\tau_d(t+1) = \tau_d(t)$. Once we get the phase rotation, we can calculate the distance change based on $d^t = \lambda \times \angle(e^{-j2\pi f_c \tau_d(t)})$, where d^t is the distance change of the dynamic path at time t, and λ is the wavelength of the audio signal. This relationship holds as long as the phase change is smaller than 2π, which holds for our finger movement speed and interval duration.

2.4.3 *Finding Channel Tap Corresponding to the Target*

Section 2.4 assumes we already know which tap to use for tracking the finger movement. This section describes how to find the right tap that includes the path reflected from the finger among multiple possible taps. Note that as mentioned in Section 2.4.1, the phase rotation by the finger movement is observed in multiple taps rather than in a single tap. Therefore, we just need to select one of these taps.

The channel taps can be classified as dynamic taps (*i.e.*, those that includes dynamic paths) and static taps (*i.e.*, those that do not). The right taps should be dynamic taps, since we are interested in tracking finger movement. If all taps are static taps, it means the finger does not move and its position does not need to be updated.

Criterion 1: Compared to the static taps, the dynamic taps have relatively larger variation of the channel over time. Therefore, we develop the following test to identify dynamic paths in the tap k: $M_1[k]^t = \frac{|h[k]^t - h[k]^{t-1}|}{|h[k]^t|} > \sigma_l$, which compares the normalized difference in the magnitude of two consecutive channels with a threshold σ_l. We use $\sigma_1 = 0.05$.

Criterion 2: While the above condition distinguishes between the dynamic and static taps, the noise in the channel estimation might cause the classification error. Therefore, we add another criterion based on the following observation: the phase rotation of static tap k, denoted as $h_d[k]$, is very unstable because all static paths are removed during the differentiation process and the remaining value in $h_d[k]$ may contain random noise. In comparison, if k is a dynamic tap, the phase rotation of $h_d[k]$ is much more stable because $h_d[k]$ includes the dynamic path and its phase change over the measurement interval is relatively small. This is evident from Figure 3 and 4, which show the phase changes when the dynamic path is not included in the channel tap.

Based on this observation, we develop the following criterion to select the final tap. We measure the stability of the phase change, which is defined as the phase change difference over the last three measurements. Specifically, we find the maximum phase change over the three periods, denoted as $M_2[k]^t = \max_{i=t-2,t-1,t} f(i)$, and $f(i) = |\angle(e^{-j2\pi f_c \tau_d^k(t)}) - \angle(e^{-j2\pi f_c \tau_d^k(t-1)})|$. We select the tap with the smallest maximum phase change (*i.e.*, the smoothest tap) from all taps that satisfy the criterion 1 as the final tap.

2.5 Estimating Absolute Distance

So far, we have focused on tracking the distance change of the finger by observing the phase. We need an absolute distance at some point in order to get the absolute distance over time. Therefore, we develop a method to estimate the absolute distance based on the channel coefficients.

How to accurately estimate the absolute distance based on the channel estimate is an open problem. [12, 35] cast this problem as non-linear optimization problems, which search for the parameters associated with each channel tap such that the sum across all taps matches the overall channel measurement. This is effective when the channel is sparse. [12, 35] show this approach achieves dm-level accuracy in WiFi. We have tried this approach in our context, and found the accuracy is poor when using acoustic signals due to many multipaths, which results in many unknowns and a severely under-constrained system.

Basic framework: We develop a new formulation to address the under-constrained problem. It is motivated by the following important observation. We do not need to reconstruct complete channel profile in order to track a moving finger. Instead, we just need to reconstruct the delay and magnitude of the path that is reflected by the finger. Since a finger is small, it is reasonable to assume only one path is reflected by the finger. Therefore, we can take the difference between the two consecutive CIR estimates, which will cancel out all static paths and reduce the unknowns to the number of the parameters associated with the path that is reflected by the finger. Recall Equation 1 models the impact of reflection on the estimated channels. We take the difference between the two consecutive CIR, which removes all static paths and only keeps the path reflected by the moving finger:

$$\begin{aligned} h_d[n] = a(&e^{-j2\pi f_c(\tau+\tau_d)}\operatorname{sinc}(n-(\tau+\tau_d)W) \\ &- e^{-j2\pi f_c\tau}\operatorname{sinc}(n-\tau W)), \end{aligned} \tag{6}$$

where a and τ are the amplitude and delay of the signal reflected from the finger, respectively [2]. Based on the measured $h_d[n]$, our goal is to find τ and a that minimize the difference between the measured and estimated CIR change, where the estimated CIR is derived from the mathematical model using Equation 6.

To further improve the accuracy, we minimize the difference between the measured and estimated CIR change over multiple consecutive CIR measurements. Our implementation considers the past 3 CIR measurements (*i.e.*, the CIR change from $t-2$ to $t-1$ and from $t-1$ to t). We still compute the coordinate every interval except that we use the most recent 3 CIR measurements to construct the optimization problem. Therefore, we can improve the accuracy while maintaining 12.5 ms tracking interval.

Putting together, we solve the following optimization problem:

$$\begin{aligned} \min_{\tau,\tau_d^{est}(i),a} : &\sum_i \sum_{n=1}^{L} \Big[h_d^{t,t+i}[n] - \\ &a(e^{-j2\pi f_c(\tau+\tau_d^{est}(i))}\operatorname{sinc}(n-(\tau+\tau_d^{est}(i))W) \\ &-e^{-j2\pi f_c\tau}\operatorname{sinc}(n-\tau W))\Big]^2 + \alpha\sum_i |\tau_d^{est}(i)-\tau_d(i)| \end{aligned} \tag{7}$$

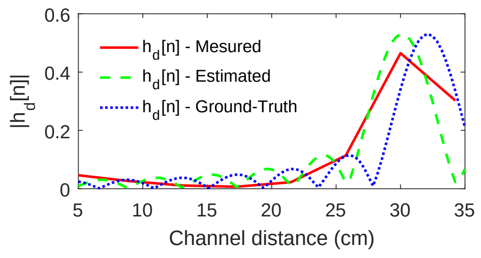
where $h_d^{t,t+i}[n]$ denotes the measured CIR change in the n-th tap from t-th measurement to $t+i$-th measurement, $\tau_d^{est}(i)$ is the inferred delay change from the t-th measurement to the $t+i$-th measurement, and $\tau_d(i)$ is the delay change derived from the phase measurement in Section 2.4. a, τ, and $\tau_d^{est}(i)$ are unknowns. The first term in the objective captures the fitting error between the measured CIR change versus the CIR change derived from the absolute distance based on a, τ and $\tau_d^{est}(i)$, and the second term captures the fitting error between the inferred delay change versus the delay change measured from the phase. α captures the relative weight between the two terms, and set to 100 in our evaluation due to more accurate distance change measurement.

Compared with the channel decomposition in [12, 35], which tries to find τ's and a's associated with all paths in the channel, our scheme finds τ and a associated with the path reflected by the moving finger. This has two benefits: (i) it reduces the number of unknowns and improves the accuracy while reducing computation cost, (ii) it removes all static paths and helps reduce the error. Moreover, we leverage the information across multiple measurements to further enhance the accuracy. Once we get τ, we can easily calculate the absolute distance as the product of delay (τ) and sound propagation speed.

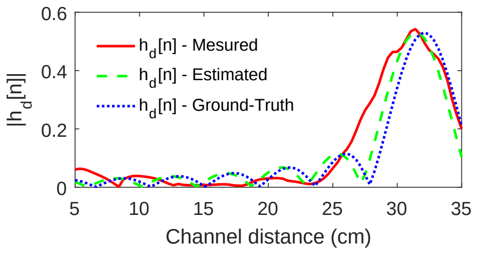
[2]Note that Equation 4 is based on Equation 3, which is an approximation to Equation 1.

Enhancement: While the above approach is useful to find the absolute distance, its accuracy is limited by the resolution of the estimated channel. Since the bandwidth of the channel is limited to 4 KHz, the baseband symbol interval is 0.25 ms, which is translated into the channel resolution of 4.25 cm. When we try to find τ using $h[n]$ sampled every 4.25 cm, the error increases due to the coarse resolution.

To enhance the accuracy, we again exploit the over-sampled signals to achieve finer resolution. For the over-sampled signals received between $h[k]$ and $h[k+1]$, we estimate the channel using the same method in Section 2.3. These samples are 0.0208 ms apart (*i.e.*, $\frac{1}{F_s}$ ms), which corresponds to the distance resolution of 3.5 mm. As a result, the resolution of the channel estimation is limited not by the baseband symbol rate but by the sampling rate of the audio signal, which is 12 times higher! Again such over-sampling is unique in our setup due to a much higher audio sampling rate than the Nyquist rate. With this fine-grained channel estimation, we find τ using the same optimization framework.



(a) Coarse grained channel



(b) Fine grained channel

Figure 5: Measured, re-generated, and ground-truth channel differences with coarse and fine grained channels based absolute distance estimation.

Figure 5 shows the impact of coarse-grained and fine-grained channel estimation on the absolute distance estimate. In this experiment, we record the audio signal while a user is moving his finger, and estimate the absolute distance using the measured channels. For ease of interpretation, we represent the x-axis of Figure 5 as the distance corresponding to the delay of the channel. The red lines in the figure show measured channel difference. The green and blue lines correspond to the channel differences by plugging in the estimated and ground-truth delays into Equation 6, respectively. The ground-truth finger distance is 32 cm. As we can see, the channel difference under the coarse channel estimation deviates from the ground-truth channel, and has 2 cm error. In comparison, the channel difference from the fine grained estimation is close to the ground-truth, and has only 0.4 cm error.

2.6 Tracking the moving finger in 2D space

In Section 2.4 – Section 2.5, we present an approach to estimate the distance to the target. This allows us to track in 1D space, which is already useful for some applications. Next we describe how to track a finger in 2D space by leveraging two microphones on a smartphone (*e.g.*, Samsung S series).

Combine relative and absolute distance: We use both absolute distance and distance change estimated from the phase to track the target. At the beginning, we use the absolute distance to get the initial position of the target. Afterwards, we can get two distance estimates: d_k^p estimated from the phase and d_k^c estimated from the channel difference, where $d_k^p = d_{k-1} + \Delta d_k^p$ and Δd_k^p is computed as a function of the phase change. We then combine the two distance estimates using a weighting factor β. β is set to 0.1 due to more accurate phase change measurement.

$$d_k = (1-\beta)d_k^p + \beta d_k^c, \tag{8}$$

Estimate coordinate: Given the phone form factor, we know the relative locations of the speaker and the microphones. Suppose the speaker is located at the origin $(0, 0)$, and the two microphones are located at $(x_1, 0)$ and $(x_2, 0)$, respectively. We assume they are all aligned in the same Y-axis because modern smartphones are very thin. The finger should be on the ellipse whose foci are $(x_1, 0)$ and $(x_2, 0)$ and the total distance to the focu is d_k. Using two microphones as landmarks, we can track the finger by finding the intersection of the two ellipses. There are two intersections between two ellipses when they overlap, and we select the one closer to the previous position.

3. PERFORMANCE EVALUATION

3.1 Experiment setup

We implement an Android app that processes the audio signal and tracks the movement of the finger in real-time. We use Samsung Galaxy S4 mobile phone as a tracking device. It has one rear speaker and two microphones at the top and bottom of the phone with 14 cm separation. The mobile app plays audio file generated as explained in Section 2.3, and receives the audio signal from the microphones, and tracks the finger position in real-time. For audio signal transmission, we use inaudible frequency band between 18 KHz and 22 KHz, and the tracking interval is 12.5 ms. To convert passband to baseband, we implement an infinite-impulse response (IIR) low-pass filter.

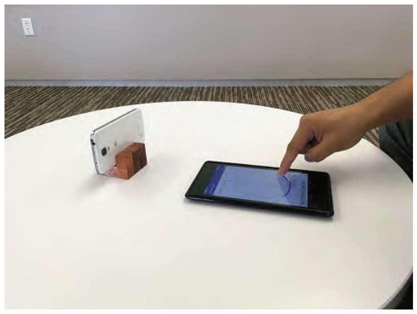


Figure 6: Testbed setup for the performance evaluation.

As shown in Figure 6, to collect the ground-truth of the finger movement, we let the user move the finger on the top of a smart tablet. It collects the touch event of the screen and generates the ground-truth trajectory of the finger movement, which is compared with the position estimated by our approach. This setup is only needed for collecting the ground truth to quantify the accuracy, and our scheme can let users freely draw in the air. In user study, we show simple shapes, such as a triangle, diamond, and circle on the tablet screen, and ask the user to trace the shapes. The average distance of the shapes is 22.3 cm. For data collection and user study, we recruit 5 people where four of them are men and one is woman. They are college students whose ages are between 21 to 29. This age-group is likely to be the most active VR/AR users.

3.2 Baseline Schemes

We compare our scheme with the following two acoustic signal based device-free tracking schemes:

Low Latency Acoustic Phase (LLAP): LLAP [29] tracks the finger movement by observing the phase change of the reflected signal. In this scheme, the transmitter continuously sends sine waves and the receiver tracks the moving target based on the phase change of the received waves. We implemented LLAP closely following [29]. We generate the transmission signal $\sin(2\pi f_c t)$ using MATLAB, stored as 16-bit PCM-format *WAV* file, and transmitted through the speaker. At the receiver side, the audio signal from the microphone is first multiplied by $\sqrt{2}\cos(2\pi f_c t)$ and $-\sqrt{2}\sin(2\pi f_c t)$, and goes through a low-pass filter to get the real and imaginary parts of the received signal, respectively. The phase of the moving finger is tracked by Local Extreme Value Detection (LEVD) algorithm in [29] that filters out static signals and tracks the phase change by the finger movement. It improves the accuracy of the phase estimation by averaging multiple received signals. Let T_s denote the sampling interval of the phase. We take the phase after averaging all received signals during T_s. T_s is selected as 12.5 ms so that both Strata and LLAP have the same tracking delay. Also, we used the multiple frequencies to address the multi-path fading as proposed in [29]. We did not implement its absolute distance estimation algorithm in LLAP for initialization, but used the ground-truth for the initial position estimation. Our evaluation results of its 1D distance estimation error is similar to what is presented in [29]. Interestingly, [29] did not evaluate the 2D trajectory error.

The main difference between Strata and LLAP is that the former separately measures the phase change of the signals with different delays while the latter measures the phase change caused by all of the surrounding objects. Since multiple body parts may move together while the user moves the finger, using the combined phase change yields inaccurate finger tracking. The problem is even worse when there are other moving objects and people nearby, which can be common in practice.

Cross-correlation based tracking: This scheme tracks a moving object by measuring the change in the cross-correlation of the received signal, called *echo profile*. It is used in FingerIO [21], which transmits an OFDM symbol every interval and locates the moving finger based on the change of the echo profiles of the two consecutive frames. Specifically, every interval it compares the difference between the echo profiles of the two consecutive frames and filters out the points whose differences are smaller than a pre-defined threshold. Among the remaining unfiltered points, it selects the point closest to the beginning of the interval since the direct path is the shortest path. We carefully followed the description of [21] in our implementation, but achieved much lower accuracy than [21] perhaps due to additional optimizations used but omitted in [21]. We identified a few ways to improve [21]. First, we use larger bandwidth and longer FFT size for OFDM symbol, both of which help to improve the tracking accuracy. We select FFT size of 256 and 6 KHz bandwidth (16 – 22 KHz) for transmission while the other approaches including ours still use 4 KHz bandwidth. Second, we find the difference between the two consecutive profiles is small as the finger is moving, which makes it challenging to select an appropriate threshold for filtering. Instead, we select the point that has the maximum cross correlation difference and our results show this helps to improve accuracy and we can now roughly track the position of the moving finger, but sometimes detect a random location due to noise. Therefore, we further filter out the positions that are too far from the previous position. Our evaluation picks the maximum peak that is within $+-$ 10 cm away from the previous position. 10 cm is a loose bound to tolerate the error in estimating the previous position. After getting the distance estimation, we use the same algorithm introduced in Section 2.6 to locate the finger in a 2D plane.

3.3 Experimental Result

3.3.1 Phase based Tracking

Tracking accuracy in 1D: We first evaluate the accuracy of estimating distance change. In this experiment, the user initially places the finger 20 cm away from the mobile phone, and moves 10 cm towards it. We repeat the experiment 200 times for each scheme and collect the CDF of the distance error. As shown in Figure 7, the median errors of Strata, LLAP, and cross-correlation based tracking (*i.e.*, CC based) are 0.3 cm, 0.7 cm, and 1.5 cm, respectively. Their 90th percentile errors are 0.8 cm, 1.8 cm, and 3.2 cm, respectively. The performance of LLAP is similar to the result presented in [29] when the finger is a moving object and the initial distance is 20 cm. Note that in [29], the authors mostly used a hand rather than a finger to evaluate the tracking performance of LLAP. Figure 7 indicates that Strata can accurately track the movement by observing the phase rotation from CIR. Its median tracking error is less than half of LLAP. Since Strata separately tracks the phase in the CIR vector, it can effectively filter out the measurement noise from the user's body movement. In comparison, LLAP cannot since all signals are mixed up. The accuracy of the cross-correlation based tracking (even after optimization) is lower than both phase-based tracking schemes.

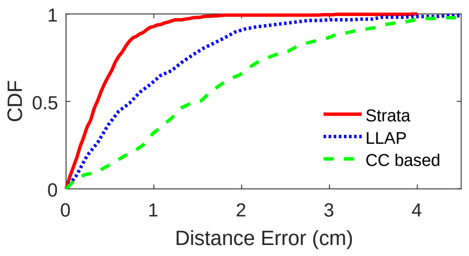


Figure 7: Impact of distance on tracking error.

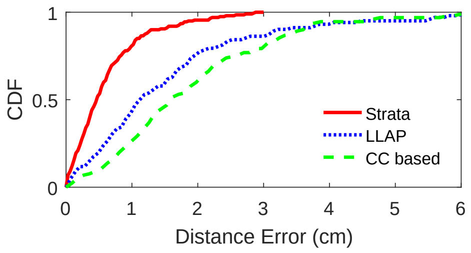


Figure 8: CDF of the distance tracking error with interrupting user.

Tracking accuracy in 1D with other moving objects: One advantage of Strata is that it can distinguish movements on the paths with different delays. This allows us to easily filter out the interference incurred by the movement of the surrounding objects in the phase tracking. By limiting the channel taps to the first 10, we can ignore the phase change caused by the moving object with the distance larger than 50 cm.

To validate its effectiveness, we perform the distance tracking experiment with a person moving in the background while another user is moving his finger. The background user is approximately 60 cm away from the mobile phone. Figure 8 plots CDF of the distance tracking error. In Strata, the tracking accuracy is almost not affected by the background user: the median error increases by only 1 mm over no background moving user.

For the cross-correlation based tracking, we set it to focus on the phase change within the range between 0 to 40 cm.

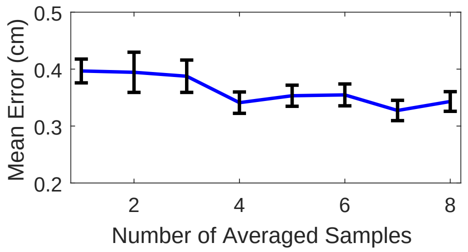


Figure 9: CDF of the distance tracking error.

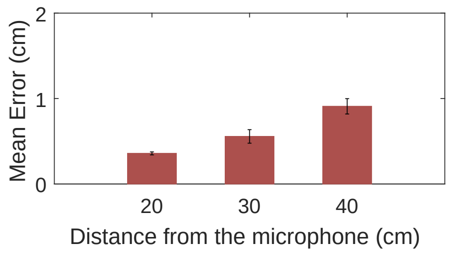


Figure 10: Impact of distance on tracking error.

Even if there is change in echo-profile in the distance of the moving user, we ignore it. As a result, it can effectively avoid the interference and achieves similar median tracking error as before: 1.6 cm.

In comparison, LLAP incurs considerable degradation due to the moving user. The median error increases from 0.8 cm to 1.2 cm. Since it does not have a mechanism to distinguish the phase change caused by multiple objects, the background movement significantly degrades its tracking accuracy.

Varying the number of samples: Figure 9 is the average 1D tracking error with 95% confidence interval when various number of samples are used for averaging downsampled signals as explained in Section 2.3. The result shows that averaging 4 samples improves the accuracy from 0.39 cm to 0.35 cm compared to simple downsampling. Moreover, it reduces the variance. The 90th percentile error decreases from 1.1 cm to 0.8 cm. Using more than 4 samples does not lead to significant additional improvement. So we use 4 samples to reduce the computation cost.

Impact of distances: We further vary the distance of the moving target. Specifically, we set the initial distance of the finger from the microphone to 20 cm, 30 cm, and 40 cm, move it 10 cm towards the mobile phone, and measure the distance error. Figure 10 shows the average error with 95% confidence interval. As the distance changes from 20 cm to 30 cm to 40 cm, the mean tracking error is 0.35 cm, 0.55 cm, and 0.9 cm, respectively. In all cases, the average error is within 1 cm. We can support a larger range by increasing the volume.

3.3.2 *Estimating Absolute Distance*

Strata can not only track the distance change from the phase, but also estimate the absolute distance using the channel difference as explained in Section 2.5. To evaluate the accuracy of the absolute distance estimation, we collected

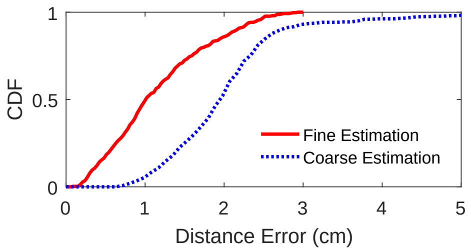


Figure 11: CDF of the absolute distance error.

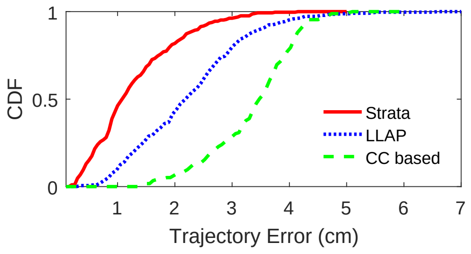


Figure 12: CDF of the trajectory error.

the audio and ground-truth data while users are tracing the shapes on the tablet where initial position of the finger is 20 cm away from the phone. The distance error is measured by calculating the difference between the estimated distance of the finger and microphones versus the ground-truth distance. Figure 11 shows the CDF of the distance error with 200 collected traces. The median and the 90th percentile errors are 1.0 cm and 2.1 cm, respectively. Note that the fine-grained channel estimation significantly improves the accuracy. In terms of the median error, the error reduction from the coarse channel estimation is 48%. We also implement the other schemes that exploits CIR to detect the distance of the target (*i.e.*, WiDeo [12] and Chronos [35]) and evaluate their absolute distance estimation accuracy, but they perform poorly for acoustic signals: their median tracking error is larger than 10 cm, so we do not include the result in Figure 11.

3.3.3 *Combining Relative and Absolute Distance Estimation*

Finally, we evaluate the tracking error in 2D plane. Given the finger's distance from the left and the right microphones, Strata and the baseline schemes track the finger position in a 2D plane as explained in Section 2.6. As shown in the previous evaluation results, the distance tracking with phase change is more reliable than the absolute distance estimation with channel difference. Therefore, we set α of Strata to 0.1 to give a higher weight to the estimate based on phase based tracking. For LLAP, we provide the ground-truth initial position and let it track the finger starting from the ground-truth initial position. We collect the finger trajectory tracked by the three schemes as well as the ground-truth trajectory while the users are following the shapes on the tablet screen. The initial distance is 20 cm. The trajectory error is calculated by averaging the difference in the distance between the estimated and ground-truth positions from all samples. Figure 12 shows the CDF of the trajectory errors of the three

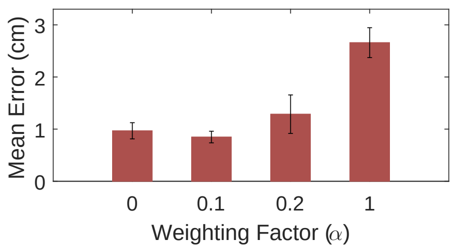
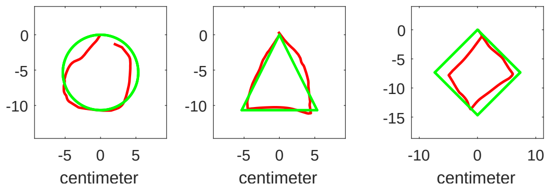


Figure 13: 2D trajectory error with various weighting factors.



(a) Circle (b) Triangle (c) Diamond

Figure 14: Median drawing error samples drawn by finger tracking.

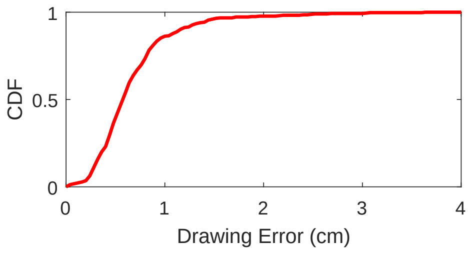


Figure 15: CDF of the drawing error in user study.

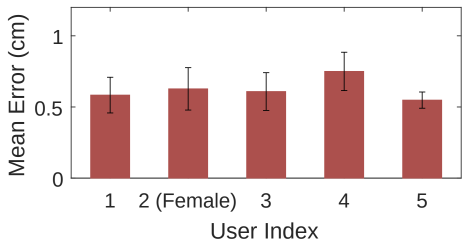


Figure 16: Mean drawing errors of 5 users.

schemes after repeating the experiment 200 times for each scheme. The median errors of Strata, LLAP, and cross-correlation based tracking are 1.08 cm, 2.18 cm, and 3.47 cm, respectively. The 90th percentile error of them are 2.40 cm, 3.50 cm, and 4.18 cm, respectively.

Figure 13 shows the average trajectory error with various α values. The result shows that when we set α to small values such as 0.1, it improves the accuracy. However, if α is higher than 0.5, it increases the error because the absolute distance error is less accurate than the phase based tracking. Note that even when $\alpha = 0$, the absolute distance estimate is useful for getting the initial position.

User study: We evaluate how accurately users can draw the shapes with real-time feedback. We implement a JAVA program that receives the tracking result from the mobile phone and visualizes a pointer on a PC screen controlled by the finger movement in real-time. Then we asked the user to move the pointer to trace the shapes (*e.g.*, triangle, diamond, and circle) on the screen. The average surface area of them are 31.1 cm. The quality of the drawings is evaluated using the drawing error, which is the same metric as the tracking error in [29] that measures the distance between original shape and the user drawing in each sampling point and averaging them. Figure 15 shows the CDF of the drawing errors using 200 trajectories collected from 5 users. The median and the 90th percentile drawing errors are 0.57 cm and 1.14 cm, respectively. When the users draw with real-time feedback, they can compensate portion of the tracking error by moving his finger towards the desired position. Figure 14 shows sample drawings of the three shapes under median drawing errors. As we can see, the traced trajectories are close to the original shapes.

Figure 16 shows the mean drawing errors of 5 users. The user index 2 corresponds to a female user while all of the other users are male. The result shows that users achieve similar accuracy.

Impact of audible noise: Since Strata uses inaudible frequency range between 18 KHz and 22 KHz, it is robust against audible noise whose frequency band is much lower than 18 KHz. During our experiment, we play music using a laptop's external speakers at their normal volume near the mobile phone, and observed similar tracking accuracy. Similar findings were reported in several different acoustic signal based tracking schemes (*e.g.*, [44, 29, 32, 18]).

Tracking time: Every 12.5ms, Strata performs low-pass filtering, passband to baseband conversion, channel estimation, and eventually tracks the finger movement. In our Samsung Galaxy S4 testbed, the average of the total processing time is 2.5 ms. According to [29], the processing time of LLAP at each interval is 4.3 ms when the same device is used. For the down-conversion, it already takes 3.5 ms. We expect that since LLAP uses multiple sine waves in different bands, during the down-conversion process, it needs to filter the signal for every band they use to receive the sine waves (9 in our implementation). On the other hand, Strata performs filtering only once for the whole band. As a result, Strata spends less than 1 ms for the down-conversion.

4. RELATED WORK

We classify existing work into (i) device-free tracking and (ii) device-based tracking.

4.1 Device-Free Tracking and Gesture Recognition

Acoustic-based tracking: Both LLAP [39] and FingerIO [21] track the finger movement using the reflected audio signal from a mobile phone. LLAP develops a phase based tracking while FingerIO uses OFDM symbol based movement detection. Our evaluation in Section 3.2 shows Strata out-performs both schemes because we extract the path associated with the finger movement and track its phase change instead of using the mixed signals [7] develops a system, called UltraHaptics, to provide haptic feedback based on acoustic radiation force from a phased array of ultrasonic transducers.

RF-based tracking: Radar uses large antenna arrays to achieve a narrow beam for tracking. It requires high-power and is heavy. WiTrack [3] applies Frequency Modulated Continuous Wave (FMCW) to WiFi to track a user's location with 10-13 cm error in x and y coordinates and 21 cm in z-dimension. In addition, since off-the-shelf radios do not perform FMCW, it implements on USRP. WiDraw [31] estimates angle of arrival (AoA) using CSI, and achieves a median tracking error of 5 cm using 25 WiFi access points (APs). The requirement of such a large number of APs significantly limits the applicability of WiDraw. In comparison, we only require 2 speakers and microphones on one machine to achieve higher accuracy. WiDeo [12] traces human motion based on reflected WiFi signals. It implements on WARP using 4 antennas and phase synchronized RX chains. Its median error is 7 cm. [46] uses 60 GHz radios and achieves cm-level accuracy for static objects with line-of-sight. [40] achieves higher tracking accuracy by using 60 GHz steerable antennas with narrow beamwidth (3.4^o) and fine-grained azimuth rotation (1^o). The major advantage of our system is high accuracy and ease of deployment since it requires no extra hardware and only software-based processing.

Vision-based: Many works use computer vision for gesture recognition and tracking. LeapMotion [16] uses sophisticated vision techniques to recognize a wide range of gestures. Kinect [1] uses depth sensors and Wii [2] uses infrared cameras to track movement. Vision approaches generally require good lighting conditions, visually distinct patterns, and high computation cost, which limit its applicability.

Device-free gesture recognition: WiSee [24] is a pioneering work that uses WiFi signals to recognize 9 gestures in several environments. AllSee [13] uses RFID tags to recognize gestures based on TV signals. [4] uses CSI from WiFi to infer which key a user types. [10, 8] use the Doppler shift of the audio signal to detect in-air gestures. ApneaApp [20] uses FMCW to track heartbeat by looking at periodic patterns. Gesture recognition performs pattern matching and requires significant training data. In comparison, continuous tracking is more challenging due to the lack of training data or patterns to match against.

4.2 Device-based Tracking

IMU-based: IMU sensors have been commonly used for motion tracking. Several works have reported that accelerometer has large error and its error increases rapidly over time due to double integration over time [38, 44]. Gyroscope has pretty good accuracy, but is not easy to use, since users have difficulty in how much to rotate in order to control certain displacement movement [44].

Audio-based: Audio is attractive for localization and tracking due to its slow propagation speed, which helps improve the accuracy. One line of research focuses on using audio for estimating distance. For example, [23] develops a novel approach that allows the sender and receiver with unknown clock offsets to measure the propagation delay. [45] addresses several practical challenges in using audio signals for tracking (*e.g.*, quickly detecting the signal, reducing computation overhead, and accounting for measurement error during movement).

Another line of research focuses on applying audio for localization. Cricket [30] uses both audio and RF to achieve median error of 12 cm with 6 beacon nodes. Swadloon [11] exploits the Doppler shift of audio signal for fine-grained indoor localization. Its error is around 50 cm. [15] uses chirp-based ranging to achieve localization error within 1 m. [44] uses the Doppler shift to estimate the velocity, based on which it computes the distance to localize the mobile. Its median error is 1.4cm.

The third line of audio based approaches target gesture recognition. [5, 32] use the Doppler shift of the audio signal between the two mobile phones for gesture recognition. DopLink [5] detects if a device is being pointed by another device. Spartacus [32] uses the Doppler shift to determine the device's moving direction and pairs it with another device moving in the same direction.

RF-based: WiFi has been widely used for localization and tracking (*e.g.*, [6, 26, 28, 35]). Many WiFi based localization uses received signal strength and their accuracy is limited due to obstacles and multipath. More recently, several works use channel state information (CSI), which reports the RSS on each OFDM subcarrier group, to provide accurate location distinction. ArrayTrack [42] and RF-IDraw [38] further use phase of the received signal to enhance the accuracy. For example, ArrayTrack achieves a median error of 23 cm using 16 antennas. RF-IDraw [38] achieves 3.7 cm tracking error.

5. CONCLUSION

This paper develops a novel device-free acoustic tracking system that achieves 1.0 cm median tracking error. Through this process, we gain several important insights: (i) phase-based tracking is effective for acoustic signals due to its small wavelength, but we need to use the channel estimate from an appropriate tap (instead of the overall channel) to achieve high accuracy, (ii) it is hard to estimate the absolute distance by directly decomposing the channel since acoustic channel is not sparse, and our formulation based on the change removes static paths, which significantly reduces the number of unknowns and improves accuracy, and (iii) acoustic channel has much higher sampling rate than Nyquist rate and our fine-grained channel estimation is effective in improving the channel estimation accuracy. Moving forward, we plan to develop applications on top of device-free tracking.

6. REFERENCES

[1] Microsoft X-box Kinect. `http://xbox.com`.

[2] Nintendo Wii. `http://www.nintendo.com/wii`.

[3] F. Adib, Z. Kabelac, D. Katabi, and R. Miller. Witrack: Motion tracking via radio reflections off the body. In *Proc. of NSDI*, 2014.

[4] K. Ali, A. X. Liu, W. Wang, and M. Shahzad. Keystroke recongition using wifi signals. In *Proc. of ACM MobiCom*, 2015.

[5] M. T. I. Aumi, S. Gupta, M. Goel, E. Larson, and S. Patel. Doplink: Using the doppler effect for multi-device interaction. In *Proc. of ACM UbiComp*, pages 583–586, 2013.

[6] P. Bahl and V. N. Padmanabhan. Radar: An in-building RF-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784, 2000.

[7] T. Carter, S. A. Seah, B. Long, B. Drinkwater, and S. Subramanian. Ultrahaptics: Multi-point mid-air haptic feedback for touch surfaces. In *Proc. of UIST*, 2013.

[8] K.-Y. Chen, D. Ashbrook, M. Goel, S.-H. Lee, and S. Patel. Airlink: Sharing files between multiple devices using in-air gestures. In *Proc. of ACM Ubicomp*, 2014.

[9] Dirac delta function. `https://en.wikipedia.org/wiki/Dirac_delta_function`.

[10] S. Gupta, D. Morris, S. Patel, and D. Tan. Soundwave: using the doppler effect to sense gestures. In *Proc. of the SIGCHI*, pages 1911–1914, 2012.

[11] W. Huang, Y. Xiong, X.-Y. Li, H. Lin, X. Mao, P. Yang, and Y. Liu. Shake and walk: Acoustic direction finding and fine-grained indoor localization using smartphones. In *Proc. of IEEE INFOCOM*, 2014.

[12] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti. Wideo: Fine-grained device-free motion tracing using RF backscatter. In *Proc. of NSDI*, 2015.

[13] B. Kellogg, V. Talla, and S. Gollakota. Bringing gesture recognition to all devices. In *Proc. of NSDI*, April 2014.

[14] S. Kumar, S. Gil, D. Katabi, and D. Rus. Accurate indoor localization with zero start-up cost. In *Proc. of ACM MobiCom*, pages 483–494, 2014.

[15] P. Lazik and A. Rowe. Indoor pseudo-ranging of mobile devices using ultrasonic chirps. In *Proc. of ACM SenSys*, pages 99–112, 2012.

[16] Leap motion. `https://www.leapmotion.com/`.

[17] J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev. Soli: ubiquitous gesture sensing with millimeter wave radar. In *Proc. of SIGGRAPH*, 2016.

[18] W. Mao, J. He, and L. Qiu. Accurate audio tracker. In *Proc. of ACM MobiCom*, Oct. 2016.

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

[20] R. Nandakumar, S. Gollakota, and N. Watson. Contactless sleep apnea detection on smartphones. In *Proc. of ACM MobiSys*, 2015.

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

[22] A. V. Oppenheim, R. W. Schafer, J. R. Buck, et al. *Discrete-time signal processing*, volume 2. Prentice-hall Englewood Cliffs, 1989.

[23] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan. BeepBeep: a high accuracy acoustic ranging system using COTS mobile devices. In *Proc. of ACM SenSys*, 2007.

[24] Q. Pu, S. Gupta, S. Gollakota, and S. Patel. Whole-home gesture recognition using wireless signals. In *Proc. of ACM MobiCom*, 2013.

[25] M. Pukkila. Channel estimation modeling. *Nokia Research Center*, 2000.

[26] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen. Zee: zero-effort crowdsourcing for indoor localization. In *Proc. of ACM MobiCom*, 2012.

[27] T. S. Rappaport et al. *Wireless communications: principles and practice*, volume 2. Prentice Hall PTR New Jersey, 1996.

[28] S. Sen, J. Lee, K.-H. Kim, and P. Congdon. Avoiding multipath to revive inbuilding wifi localization. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 249–262. ACM, 2013.

[29] D.-F. G. T. U. A. Signals. Wei wang and alex x. liu and ke sun. In *Proc. of MobiCom*, Oct. 2016.

[30] A. Smith, H. Balakrishnan, M. Goraczko, and N. Priyantha. Tracking moving devices with the cricket location system. In *Proc. of ACM MobiSys*, 2005.

[31] L. Sun, S. Sen, D. Koutsonikolas, and K. Kim. Widraw: Enabling hands-free drawing in the air on commodity wifi devices. In *Proc. of ACM MobiCom*, 2015.

[32] Z. Sun, A. Purohit, R. Bose, and P. Zhang. Spartacus: spatially-aware interaction for mobile devices through energy-efficient audio sensing. In *Proc. of ACM Mobisys*, pages 263–276, 2013.

[33] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, A. Topalian, E. Wood, S. Khamis, P. Kohli,

S. Izadi, R. Banks, A. Fitzgibbon, and J. Shotton. Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences. In *Proc. of SIGGRAPH*, 2016.

[34] D. Tse and P. Viswanath. *Fundamentals of wireless communication*. Cambridge university press, 2005.

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

[36] Vive. `http://www.htcvive.com`.

[37] J. Wang, F. Adib, R. Knepper, D. Katabi, and D. Rus. RF-compass: robot object manipulation using RFIDs. In *Proc. of the 19th annual international conference on Mobile computing & networking*, pages 3–14, 2013.

[38] J. Wang, D. Vasisht, and D. Katabi. RF-IDraw: virtual touch screen in the air using RF signals. In *Proc. of ACM SIGCOMM*, 2014.

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

[40] T. Wei and X. Zhang. mTrack: high precision passive tracking using millimeter wave radios. In *Proc. of ACM MobiCom*, 2015.

[41] Y. Xie, Z. Li, and M. Li. Precise power delay profiling with commodity wifi. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 53–64. ACM, 2015.

[42] J. Xiong and K. Jamieson. Arraytrack: A fine-grained indoor location system. In *Proc. of NSDI*, pages 71–84, 2013.

[43] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu. Tagoram: Real-time tracking of mobile RFID tags to high precision using cots devices. In *Proc. of ACM MobiCom*, 2014.

[44] S. Yun, Y. chao Chen, and L. Qiu. Turning a mobile device into a mouse in the air. In *Proc. of ACM MobiSys*, May 2015.

[45] Z. Zhang, D. Chu, X. Chen, and T. Moscibroda. Swordfight: Enabling a new class of phone-to-phone action games on commodity phones. In *Proc. of ACM MobiSys*, 2012.

[46] Y. Zhu, Y. Zhu, B. Y. Zhao, and H. Zheng. Reusing 60 GHz radios for mobile radar imaging. In *Proc. of ACM MobiCom*, 2015.