EarphoneTrack: Involving Earphones into the Ecosystem of Acoustic Motion Tracking

Gaoshuai Cao*
University of Science and Technology of China, Hefei, China
cgs@mail.ustc.edu.cn

Kuang Yuan*
University of Science and Technology of China, Hefei, China
kelleykuang@mail.ustc.edu.cn

Jie Xiong
University of Massachusetts Amherst
Massachusetts, USA
jxiong@cs.umass.edu

Panlong Yang
University of Science and Technology of China, Hefei, China
Peng Cheng Laboratory
plyang@ustc.edu.cn

Yubo Yan
Hao Zhou
University of Science and Technology of China, Hefei, China
{yobuyan,kitewind}@ustc.edu.cn

Xiang-Yang Li
University of Science and Technology of China, Hefei, China
xiangyangli@ustc.edu.cn

ABSTRACT

Acoustic motion tracking is an exciting new research area with promising progress in the last few years. Due to the inherent low propagation speed in the air, acoustic signals have the unique advantage of fine sensing granularity compared to RF signals. Speakers and microphones nowadays are pervasively available in devices surrounding us, such as smartphones and voice-controlled smart speakers. Though promising, one fundamental issue hindering the adoption of acoustic-based motion tracking is that the positions of microphones and speakers inside a device are fixed, which greatly limits the flexibility of acoustic motion tracking. In this work, we propose a new modality of acoustic motion tracking using earphones. Earphone-based tracking mitigates the constraints associated with traditional smartphone-based tracking. With novel designs and comprehensive experiments, we show earphone-based motion tracking can achieve a great flexibility and a high accuracy at the same time. We believe this is an important step towards "earable" sensing.

CCS CONCEPTS

• **Human-centered computing** → *Sound-based input / output*.

KEYWORDS

Earable sensing, Earphone-based acoustic sensing, Motion tracking

ACM Reference Format:
Gaoshuai Cao, Kuang Yuan, Jie Xiong, Panlong Yang, Yubo Yan, Hao Zhou, and Xiang-Yang Li. 2020. *EarphoneTrack*: Involving Earphones into the Ecosystem of Acoustic Motion Tracking. In *The 18th ACM Conference on Embedded Networked Sensor Systems (SenSys '20), November 16–19, 2020, Virtual Event, Japan.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3384419.3430730

*Gaoshuai Cao and Kuang Yuan are co-first authors

SenSys '20, November 16–19, 2020, Virtual Event, Japan

ACM ISBN 978-1-4503-7590-0/20/11…$15.00
https://doi.org/10.1145/3384419.3430730

1 INTRODUCTION

Wireless sensing has been a hot research topic in recent years with a large range of applications enabled including tracking [33, 40, 46, 62], gesture recognition [10, 42, 49] and vital sign monitoring [52, 59]. Among the wireless signals employed for sensing, acoustic signals exhibit unique advantages in granularity and accuracy due to the extremely low signal propagation speed in the air (340 m/s). Recent efforts have pushed the sensing accuracy to millimeter-level [31, 50]. On the other hand, speakers and microphones are becoming essential components in a lot of daily devices, such as smartphones, personal computers, smart TVs, and smartspeakers (*e.g.*, Amazon Alexa [4] and Google Home [22]).

Promising progress has been achieved in acoustic motion tracking and the proposed systems can be broadly divided into two categories: device-free [36, 45, 54, 61] and device-based [33, 50, 60, 63, 64] Take hand tracking as the example. Device-based tracking requires the user to hold a device in hand and the hand motion is captured by tracking the device in the hand. On the other hand, in device-free tracking, both microphone and speaker are kept static and the signal reflected from the hand is employed to track the hand movement. Among these systems, the majority are hosted on smartphones due to pervasive smartphone usage. For device-free tracking based-on smartphones, due to the intrinsic nature of relying on weak reflection signals for tracking, the tracking range is usually limited to less than one meter. On the other hand, device-based approaches usually have a larger tracking range. Though promising, several severe issues are associated with acoustic motion tracking with a smartphone.

The first issue is that the microphones and speakers built in a smartphone have fixed positions which greatly limits the tracking flexibility and capability. For device-free motion tracking, the tracking area is confined to be very close to the phone and motion tracking usually only works at the up and down sides but not the left and right sides of the phone. To mitigate the issues, a lot of systems employ Arduino [6] or Bela [8] platforms which have some degree of freedom to vary the position and number of the microphone/speakers. However, even for these platforms designed for flexibility, the freedom is still limited. The microphones and speakers need to be either directly connected to the platform, thus restricted within a small area or connected with messy wires.

(a) A new HCI tool for smart apps. (b) A flexible contactless tracking. (c) A 'touchable' screen . (d) A small VR controller.

Figure 1: Application scenarios of earphone-based motion tracking.

The second issue is that the smartphone is still too big to be held in hand for device-based tracking. In [2, 50, 60], researchers proposed to write in the air using a smartphone as a "big" pen. However, people fatigue easily to write in the air using a smartphone. Furthermore, for fine-grained tracking, the large phone size also limits the tracking accuracy.

The third issue is that in device-based tracking, the smartphone is the tracking target (e.g., held in hand and write in the air), the display function of the smartphone disappears. During the tracking process, the smartphone may constantly move, preventing the screen from being clearly visualized. In the VR application, the smartphone needs to be tucked into the VR glasses, preventing it from being held in hand for tracking purposes.

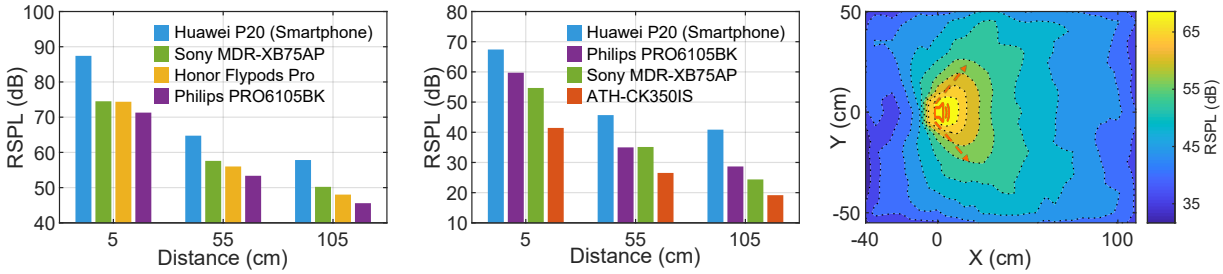
In this paper, we present *EarphoneTrack*, an earphone-based acoustic motion tracking system. Commercial earphones, including both wireless and wired earphones, are now becoming more powerful and popular. When these earphones are not used for music play and phone call, we observe exciting opportunities to employ them to enable new wireless sensing applications. We therefore propose to include earphones into the ecosystem of acoustic motion tracking. With small and lightweight earphones, we could extend the tracking area from a small region in front of the phone to meters away with great flexibility as shown in Fig. 1b. For device-based motion tracking, it is also much more convenient to hold a small earbud in hand compared to holding a phone. With two earbuds, we can track two hands simultaneously. With increasing popularity of wireless earphones such as Airpods Pro, we believe the proposed earphone-based motion tracking system has a great potential to enable exciting new applications. We believe this is an important step towards the era of "earable" sensing.

The basic idea sounds straightforward. However, it is non-trivial to realize *EarphoneTrack* due to the following challenges:

- *Strong self-interference in wired earphone.* Speaker and microphone in a wired earphone are separated by insulating material to mitigate interference so people can listen and speak at the same time. In reality, the insulating layer does not work perfectly and there is still a signal leakage from the speaker to microphone. This leakage is very small and is not an issue for everyday use such as phone calls. However, it becomes an issue when we employ it for motion tracking. For everyday use, the volume (signal amplitude) of the earphone is usually tuned to 25% of the maximum value. However, in motion tracking, the maximum signal amplitude is adopted. Furthermore, for motion tracking, the inaudible frequency band between 16 kHz and 22 kHz is used and this frequency is much higher than that of human voice (0.5 kHz - 3 kHz). The amount of leakage is related to signal frequency and amplitude. The bad news is that both high frequency and large volume increase the amount of leakage and thus the leakage in motion tracking is tens of times larger than that in everyday use, interfering with the received signal.
- *Large frequency offset in wireless earphone.* Accurate phase measurement is the key for motion tracking. For wireless earphone, we find that there exists a large frequency offset between the expected signal and the actually generated signal. For most smartphones, the built-in oscillator is able to generate a signal with a frequency offset less than 0.001 Hz. This small frequency offset is negligible for motion tracking. However, the frequency offset for wireless (Bluetooth) earphone is usually greater than 0.15 Hz due to the very small oscillator adopted. For AirPods 2, the offset is as large as 1 Hz. This large phase offset leads to a large error in motion tracking.
- *Narrow bandwidth.* Existing acoustic motion tracking systems employ Frequency Modulated Continuous Wave (FMCW) chirp signal for accurate range estimate. The tracking performance is linearly related to the bandwidth of the chirp signal. For smartphone, the inaudible band which can be utilized for tracking is around 6 kHz (16 kHz - 22 kHz). This is enough to support highly accurate estimates. However, this large inaudible band does not exist at most wireless earphones. Samsung Galaxy Buds+ is able to support only 1 kHz (16 kHz - 17 kHz) band while Apple Airpod 2 can only support around 0.5 kHz (16 kHz - 16.5 kHz) band. Thus, traditional chirp-based signal design does not work well for earphone-based motion tracking.

To tackle the first challenge, we deeply analyze and model the relationship between the transmitted signal and leaked signal. We then propose a simple but efficient scheme to eliminate the leaked interference signal for motion tracking.

To address the issue of frequency offset at wireless earphones, we propose to compensate the frequency offset induced distance deviation. If both transceivers are static, the measured distance is a constant when there is no frequency offset. On the other hand, the distance varies with time if there exists a frequency offset and the distance varies linearly if the frequency offset is a constant. With comprehensive experiments, we find that this frequency offset does not change within a power on/off cycle of the hardware. We therefore compensate the frequency offset induced distance deviation to achieve accurate tracking.



(a) Signal strengths of different speakers. **(b) Signal reception capability of microphones.** **(c) Radiation pattern of an earphone speaker**

Figure 2: Benchmark experiments on earphone acoustic components.

To deal with the last challenge, we adopt a single-frequency sinusoidal signal rather than a chirp signal for earphone-based motion tracking. We propose a novel peak/valley based distance measurement scheme for fine-grained tracking. We employ a time window that contains a fixed number of peaks and valleys for distance measurement. The basic idea is that when there is no relative movement between the transceiver pair, the time taken to receive the transmitted sinusoidal signal is exactly the same as the time taken to transmit the signal. When the receiver moves towards the transmitter, the time taken to receive the transmitted signal will be shorter than the time taken to transmit the signal. In this case, the received sinusoidal signal looks like being compressed in a smaller time window compared to the transmitted signal. Similarly, when the receiver moves away from the transmitter, the signal is like being stretched. To accurately calculate how much the signal is compressed/stretched for distance measurement, we employ a window with a fixed number of peaks and valleys to calculate the phase difference between the transceivers and accordingly convert this phase difference to distance measurement. To further improve the accuracy, we up-sample the received signal by zero padding and low-pass filtering [3].

Summary of Results. We implement the device-based acoustic motion tracking system on both smartphone and PC platforms and evaluate the performance using multiple commercial earphones of different brands including Apple, Samsung and Sony. The proposed earable motion tracking system is able to achieve *mm*-level accuracy. For 1D, 2D and 3D device-based tracking, *EarphoneTrack* achieves a tracking accuracy of 1.1 mm, 1.9 mm and 6.9 mm, respectively. The results show that the proposed earphone tracking systems achieve a great flexibility without sacrificing the tracking accuracy. The end-to-end system latency is around 5 ms, which is small enough for real-time tracking. You can find the demo video of one 2D tracking example of our system at: https://youtu.be/3VhBBxCABZ0.

Contribution. To summarize, we made the following main contributions in this work:

(1) We include commercial earphone into the ecosystem of acoustic motion tracking to address several limitations of exiting smartphone-based acoustic motion tracking.
(2) We identify unique challenges associated with earphone motion tracking and propose solutions to address them, enabling motion tracking using both wired and wireless earphones.
(3) We implement *EarphoneTrack* on both Android and Linux platforms. Comprehensive experiments demonstrate the feasibility and advantages of employing earphones for motion tracking. We believe this is an important step towards earable sensing.

2 BACKGROUND ON EARPHONES

In this section, we introduce the basics of commercial earphones. We present the earphone's internal structure and explain the underlying mechanisms of the strong self-interference and the frequency offset associated with earphone-based motion tracking.

2.1 Earphone basics

An earphone is an audio device where the electrical and acoustic signals are converted to each other. As shown in Fig. 3, it contains two key components: speaker and microphone. The speaker translates the electric signals into a corresponding acoustic signal. Specifically, the electric current exerts a varying force on the diaphragm, causing it to vibrate, creating sound waves. On the contrary, sound waves hit a diaphragm to make it vibrate and this vibration is converted into an electrical signal either through a capacitor or a coil.

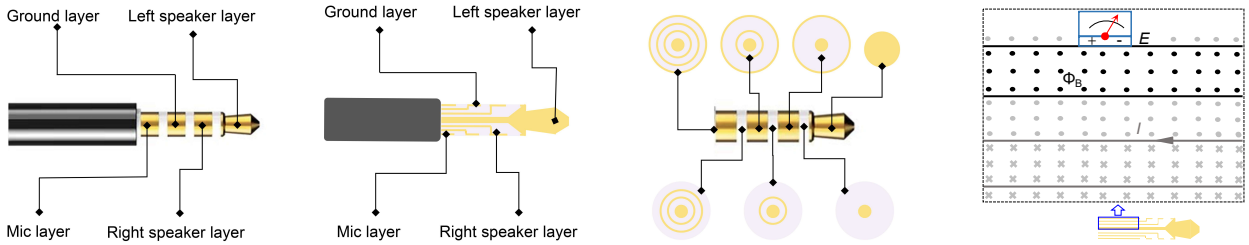
In addition to speakers and microphones, for wired earphone, a plug is used to connect it to devices such as a smartphone. This plug is replaced with a Bluetooth module for wireless earphone. More advanced earphones even integrate a variety of sensors into them. For example, accelerometer is integrated in AirPods Pro [23] and touch sensor is embedded in Sony WF-1000XM3 [20].

(a) A wired earphone [7]. **(b) A wireless earphone [5].**

Figure 3: Acoustic components of an earphone

2.2 The signals transmitted from and received at the earphone are weaker

We conduct benchmark experiments to compare the signal transmission and reception capabilities of an earphone compared with a smartphone. We employ an acoustic signal with a frequency in the range of 16 kHz - 17 kHz, which is supported by most commercial earphones [24, 26, 27] and is also inaudible for most people [29]. We employ the Relative Sound Pressure Level (RSPL) as the metric to measure the sound intensity defined as below:



(a) An earphone plug. (b) Longitudinal section view. (c) Cross section view. (d) Interfering signal E appears when I in the speaker circuit changes.

Figure 4: The internal structure of a earphone plug

$$RSPL_{dB}(x) = 20\log_{10}(A_x) \quad (1)$$

where A_x is the amplitude of the received sound signal when the distance between the microphone and the speaker is x. In the first experiment, we employ two wired earphones (Sony MDR-XB75AP [19] and Philips PRO6105BK [17]), one wireless earphone (Honor Flypods Pro) and the built-in speaker of a smartphone (Huawei P20 [14]) to transmit a single-frequency acoustic signal at 16 kHz. The volume is set to the maximum value. The receiver is the built-in microphone of another smartphone (Huawei P20 [14]). The distance between the speaker and the microphone is increased from 5 cm to 105 cm. At each location, we collect sound signal for 5 seconds. The amplitudes of the received signals are shown in Fig. 2a. We observe that the sound signal transmitted from an earphone speaker is about 5 dB - 20 dB weaker compared to the sound signal transmitted from the built-in speaker of a smartphone. We observe similar results with other smartphones serving as the transmitter including Samsung Galaxy S7, Huawei Nexus 6p and iPhone 11 Pro.

In the second experiment, we send a 16 kHz sound signal from a smartphone and employ four different microphones (Huawei P20, Philips PRO6105BK, Sony MDR-XB75AP and ATH-CK350IS [11]) to receive the signal at a same distance. Fig. 2b shows that the amplitudes of the received signals at the earphone microphones are 5 dB - 25 dB smaller than that of the signal received at the microphone built in the Huawei smartphone.

Compared to the smartphone speaker, people may think the sound signal emitted from an earphone speaker is more directional. In the third experiment, we measure the signal strength at different positions when an earphone speaker (Moshi Mythro [16]) transmits sound signals. The volume is tuned to its maximum value. The signal strength heatmap is plotted in Fig. 2c. We can see that the sound beam has a width around 100° which is comparable to the smartphone speaker.

2.3 Signal leakage associated with wired earphone

In the previous experiments, the speaker and microphone are on two separate devices. For motion tracking, we usually just use one device and employ the speaker and microphone on the same earphone to transmit and receive signals. We find that there exists a strong signal leakage causing severe interference when a wired earphone is utilized for motion tracking.

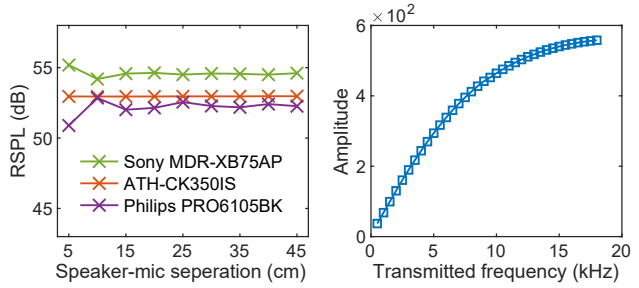
We conduct one experiment to show the effect of this leakage. We employ the same earphone to transmit and receive signals. Three different wired earphones are employed in this experiment. We increase the distance between the speaker and the microphone and measure the strength of the signal received. As shown in Fig. 5a, when the distance is increased from 5 cm to 45 cm, the strength of the received signal does not decrease but surprisingly remains almost the same. We carefully study this phenomenon and find this is due to signal leakage. Besides the signal propagated through the air and received at the microphone, there is a leakage through the plug from the transmitter to the receiver. What makes it worse is that this leakage signal is much stronger than the received signal through the air and thus when the two signals are mixed, the leakage signal dominates. Therefore, even though the received signal through the air varies with the transmitter-receiver distance, we are not able to see the effect as the dominating leakage signal does not change.

Now we explain why this leakage occurs. We start from the internal structure of the earphone plug. As shown in Fig. 4a, the plug of a wired earphone contains four metal layers. The four layers are connected to the microphone, ground, speaker right channel, speaker left channel, respectively. Adjacent layers are separated by an insulating layer. The view of longitudinal slice is shown in Fig. 4b and it contains the circuit structure shown in Fig. 4d. The circuit structure is composed of four parallel circuits, corresponding to the metal layer, connected to the microphone, ground, speaker right channel, and speaker left channel from top to bottom. When there is a changing current I in the circuit connected to the speaker left or right channel, it will cause changes of the magnetic flux Φ_B in the closed loop, which consists of the microphone circuits and the ground circuits, and then generate an induced electromotive force E in the microphone circuit. In our context, to generate a sound signal of a specific frequency at the speaker, an eclectic current of the same frequency $I = A\sin(2\pi ft)$ is created in the circuit. The strength of the magnetic field induced by the current flow at a distance of r is expressed as below [30]:

$$\boldsymbol{B} = \frac{\mu_0 I}{2\pi r} = \frac{A\mu_0 \sin(2\pi ft)}{2\pi r} \quad (2)$$

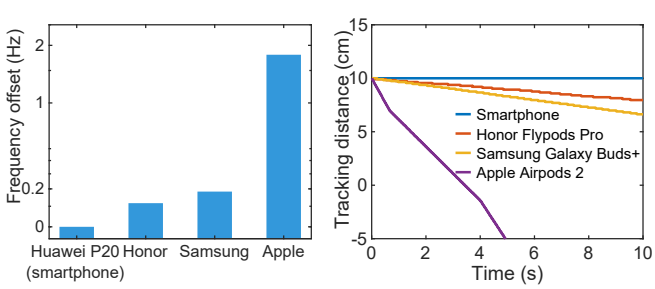
where μ_0 is the vacuum permeability, A is the current amplitude and f is the current frequency. From Maxwell's equations, the induced electromotive force E is calculated as the changing speed of the the magnetic flux:

$$\begin{aligned} \boldsymbol{E} &= -\frac{d}{dt}(\boldsymbol{B}\cdot S) \\ &= -\frac{A\mu_0}{2\pi r}\cdot\frac{d}{dt}(\sin(2\pi ft)\cdot S) \\ &= -A\cdot S\cdot f\cdot\frac{\mu_0}{r}\cos(2\pi ft) \end{aligned} \quad (3)$$



(a) RSPLs when microphone and speaker share the same plug. **(b) Amplitude of leaked signal**

Figure 5: Self interference



(a) Frequency offset of different devices **(b) Distance shift vs. Time**

Figure 6: Frequency offset

where S is the area the magnetic field goes through. From Eq. 3, we can see that the amplitude of the leaked signal is not just linearly related to the original signal's amplitude but also the the frequency. The higher frequency, the stronger leakage. Compared to MUSIC play and phone call, we employ a much higher frequency signal for motion tracking. Therefore, this leakage issue is more severe. We measure the signal leakage by increasing the signal frequency and the results are plotted in Fig. 5b. The strength of the leaked signal increases with frequency as expected. However, we observe when the frequency approaches 15 kHz, the leakage gradually saturates. This is because when the signal frequency increases, the impedance also increases, attenuating the signal leakage.

2.4 Frequency Offset in wireless earphones

Due to the small size of earphones, the oscillator adopted is not as accurate as that used in smartphones. Therefore, there is a frequency offset between the expected signal and the actually generated signal. A small frequency offset has almost no effect on music listening but it is a big issue for fine-grained motion tracking. For a 16 kHz signal, a 1 Hz frequency offset can cause a tracking error of 2.125 cm ($\frac{\Delta f}{f}ct$) in one second. We find that most of the smartphones and wired earphones have very small frequency offsets which can be neglected. However, the frequency offsets of wireless earphones are significantly larger. We measure the frequency offsets when a 16 kHz signal is transmitted using three wireless earphones as well as a smartphone. The results are shown in Fig. 6a. We can see that the frequency offset of the smartphone is very close to 0. On the other hand, the frequency offsets of Honor Flypods Pro and Samsung Galaxy Buds+ are in the range of 0.1 Hz - 0.2 Hz. Surprisingly, Apple Airpods 2 has a much larger frequency offset which is around 1.8 Hz. We also notice that the frequency offset is a fixed value in each power on/off cycle. This means as long as we do not power restart the hardware, the frequency offset does not change. This observation is also demonstrated in 6b. We can see strictly linear tracking errors with respect to time for Samsung and Honor earphones, indicating constant phase offsets. The slope corresponds to the magnitude of the frequency offset and Apple earphone has a much larger frequency offset.

3 SYSTEM OVERVIEW

EarphoneTrack is an earphone-based motion tracking system. It employs commercial earphones to transmit and receive inaudible acoustic signals for motion tracking without any hardware modification. Band-pass filters are used to remove the background noise and also separate received signals from different speakers transmitting using different frequencies. To accurately measure the distance using an earphone, strong leakage needs to be removed for wired earphones. Different from smartphone-based acoustic tracking which employs chirp-based signal, a single frequency sinusoidal signal is employed in earphone tracking. For wireless earphone, the frequency offset induced range deviation needs to be compensated to obtain accurate distance measurements. By combining the distance measurements and the geometric relationships between the microphone and the speakers, *EarphoneTrack* is capable of tracking target motions in 3D at a high accuracy. Fig. 7 shows the overall system architecture consisting of four modules: i) Self-interference (leakage) cancellation; ii) Distance measurement; iii) Frequency offset compensation and iv) 1D/2D/3D target tracking.

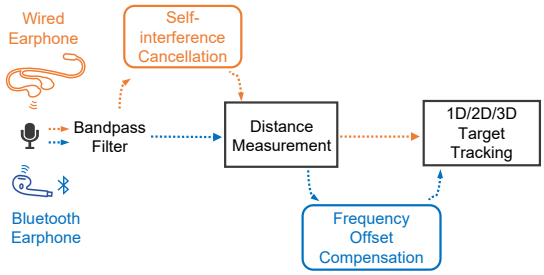


Figure 7: System architecture of *EarphoneTrack*.

4 SELF-INTERFERENCE CANCELLATION

The received signal $S_r(t)$ consists of two parts. One part is through air propagation $S_a(t)$ and the other part is the leakage $S_l(t)$ as described in Sec. 2.3. Only the air-propagation part contains the target tracking information and the leakage needs to be removed. Based on the theoretical analysis described in Sec. 2.3, we know that the leaked signal has the same frequency as the transmitted signal but a shifted phase and an amplitude attenuation. If the phase shift and amplitude attenuation are constants, we can measure them beforehand and estimate the leakage based on the frequency of transmitted signal.

4.1 Measuring signal amplitude and phase

Each sample of the received signal contains two pieces of information: phase and amplitude. We can thus express the signal sample as $V = A\sin\phi$, where ϕ is the phase and A is the amplitude. However, what is retrieved from the audio device is just V and we do not know

ϕ and A. To obtain the value for ϕ and A, we need another equation containing these two variables. For this purpose, we get the 90° shift of the received signal, $V' = A\cos\phi$, through Hilbert Transform [39]. The amplitude A can then be calculated as:

$$A = \sqrt{(A\sin\phi)^2 + (A\cos\phi)^2} \tag{4}$$

and ϕ calculated as:

$$\phi = \arctan\left(\frac{A\sin\phi}{A\cos\phi}\right) \tag{5}$$

4.2 Eliminating the leaked signal

When the transmitted signal is a single-frequency sinusoidal signal, the leaked signal can be expressed as:

$$S_l(t) = A'\sin\left(2\pi f t + \phi'\right) \tag{6}$$

where A' is the amplitude, ϕ' is the phase shift with respect to the transmitted signal and f is the signal frequency which is the same as the that of the transmitted signal. To measure the amplitude and phase information of the leaked signal, we keep the distance between the microphone and the speaker as far as possible to minimize the strength of the signal arriving at the microphone through the air. We further employ soundproof material to attenuate the signal propagation through the air. With these two measures, the signal component through the air $S_a(t)$ is small enough to be neglected and the received signal is just the leakage component. By applying Equation 4, we can obtain the amplitude A' of the leaked signal and estimate the phase shift ϕ' by comparing the difference between the phase of the leaked signal and the phase of the transmitted signal. The signal leakage is quite stable and thus this measurement process is a one-time effort. Once the leakage is measured, we can remove it from the received signals to obtain clean $S_a(t)$ for tracking. Fig. 8 shows the remaining clean signal after eliminating the leakage.

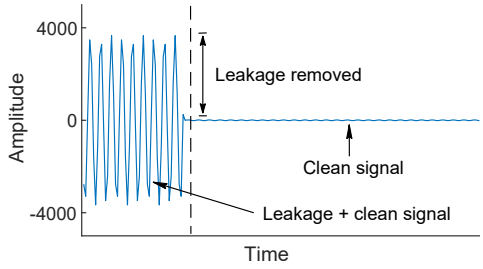


Figure 8: The clean signal after leakage elimination.

5 MOVEMENT DISTANCE MEASUREMENT

5.1 Phase-based Distance Measurement

In wireless communication, the Doppler effect is a well-known phenomenon where the frequency of the received signal slightly changes when the transmitter/receiver moves [34]. Based on the frequency change, we can obtain the speed of the receiver relative to the transmitter as:

$$v = \frac{\Delta f}{f}c \tag{7}$$

where f denotes the frequency of the transmitted signal, Δf is the frequency shift due to the Doppler Effect, and c is the signal propagation speed in air. In theory, v can be estimated by accurately measuring the frequency shift Δf. Unfortunately, it is not easy to accurately estimate Δf to meet the requirement for tracking using traditional frequency analysis schemes such as Fast Fourier Transform (FFT). FFT can present us the average frequency within a time window but it cannot capture the instantaneous frequency at each timestamp. Thus, the FFT-based schemes only offer coarse-grained tracking accuracy [42, 60]. For fine-grained tracking, existing systems [31, 54, 61, 63] mostly measure the phase change, and convert the phase change into the moving distance:

$$\Delta d = \frac{\Delta\phi}{2\pi}\lambda \tag{8}$$

A phase change of 2π corresponds to a distance change of one wavelength λ ($\lambda = 2$ cm for a 16 kHz acoustic signal). With a phase resolution of 0.1π, the distance estimation resolution is 1 mm, which is fine enough for most tracking applications.

5.2 Compression & stretching of signal waves

As shown in Fig. 9a, the speaker transmits a sinusoidal signal and the microphone receives the signal. When both transmitter and receiver are static, the time period taken for signal transmission at the transmitter is exactly the same as the time period taken for signal reception at the receiver. If the receiver remains static and the transmitter moves towards the receiver as shown in Fig. 9b, the time period taken for signal transmission is larger than the time period taken for signal reception. In this case, if we compare the received signal with the transmitted signal, the received sinusoidal wave is compressed because the same signal is now contained in a smaller time window. The transmitter displacement (Δd) can be calculated as $(t_1 - t_2) \times c$ where c is the signal propagation speed in the air. In contrast, if the transmitter moves away from the receiver, the time period taken for reception is larger than that for transmission as shown in Fig. 9c. In this case, the received signal is stretched compared to the transmitted signal.

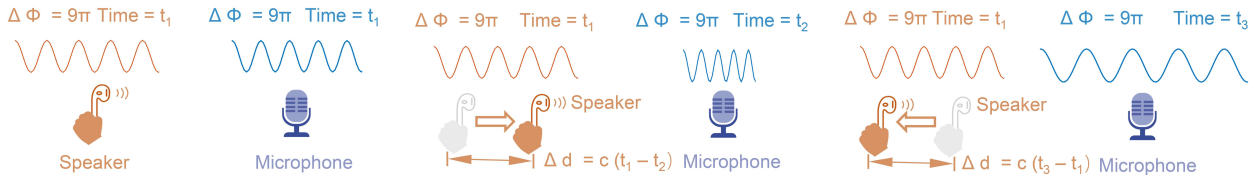
5.3 Dynamic Time Window based on the number of local extreme points

As we employ a single-frequency signal (sinusoidal wave) in our design, the phases of the transmitted signal and received signal are both time variant. Therefore, the phase change ($\Delta\phi$) in Eq. 8 is calculated as the difference of the phase change at the received signal and the transmitted signal:

$$\Delta\phi = \Delta\phi_r - \Delta\phi_t \tag{9}$$

Note that the phase change of the transmitted signal $\Delta\phi_t$ can be easily obtained by multiplying the carrier frequency with the time interval. What needs to be estimated is the phase change of the received signal $\Delta\phi_r$. $\Delta\phi_r$ can be calculated as the phase difference of the received signal samples at the beginning and ending of the time window as $\Delta\phi_r = \phi_i - \phi_j$. When transmitter or receiver moves, the length of the LoS path changes and accordingly the phase difference $\Delta\phi_r$ of the received LoS signal varies.

Most of the acoustic tracking systems [31, 54, 63] adopt Fixed Time Window (FTW) containing a fixed number of sample points as the smallest data segment to calculate phase change. In order to obtain the accumulative phase change $\Delta\phi_r$ between the starting and ending sample points, FTW solutions need to calculate the phase



(a) The transmission time and reception time of the same signal are the same when the speaker and microphone are relatively stationary.

(b) The time required to receive the signal is compressed when the speaker moves towards the microphone.

(c) The time required to receive the signal is stretched when the speaker moves away from the microphone.

Figure 9: Relationship between moving distance and signal compression or stretching

values of each sample points and unwrap them. In our design, the transmitted signal is a single-frequency sinusoidal signal. To measure the phase value of each sample point, existing approaches first obtain the In-phase (I) and Quadrature (Q) components of the signal and then calculate the phase ϕ by arctan (I/Q). Such processing either needs to obtain the 90° shift of the received signal using Hilbert Transform, or needs to multiply the received signal with a given signal and pass the mixed signal through a low-pass filter [54], which induces a relative high computational overhead.

For sinusoidal wave, the phase change is π between two adjacent local extreme points (i.e., peak and valley). We only use the phase difference between the starting and ending sample points for displacement calculation. Based on this, we employ the Dynamic Time Window (DTW) scheme to calculate the movement distance of each segment by making sure the beginning and ending sample points of the window are both local extreme points. Compared with FTW, our DTW scheme selects a data segment containing a fixed number of local extreme points rather than a fixed number of sample points to calculate the phase change. If we assume each window contains L local extreme points, the phase change of the signal of each window can be simply calculated as:

$$\Delta\phi_r = L\pi \tag{10}$$

The computational cost of this calculation is very low compared with the traditional phase measurement method based on FTW because we do not need to process the signal with Hilbert transform and unwrapping but just counting the number of local extreme points.

As explained in Sec. 5.2, the signal wave can be compressed or stretched based on the relative movement direction of the transceiver pair. When the signal wave is compressed, for the same amount of local extreme points, the time window is smaller, indicating a decreasing distance between the transceiver pair.

5.4 Peak/Valley-based Distance Measurement

When there is a relative movement between the transmitter and receiver, the sinusoidal signal received will be compressed or stretched. By measuring how much the signal wave is compressed or stretched, we can obtain the moving distance. Specifically, the movement distance (displacement) can be calculated as the difference of the time taken for signal transmission and reception multiplied by the signal propagation speed. However, it is non-trivial to accurately obtain the time difference. In this section, we present our peak/valley-based method to obtain the phase difference.

We estimate the phase change of the received signal $\Delta\phi_r$ in a window containing L peak/valleys using Eq.10. However, the sampling rates of commercial devices are not high enough to make sure the starting/ending samples are close enough to the local extreme points. For example, with a typical 48 kHz sampling rate, we only have three sample points per cycle for 16 kHz signals. In this case, the starting/ending samples can be far away from the local extreme points, and there exists a big error if we apply Eq. 10 for phase calculation at the receiver. To make the sample points close to the local extreme points, we upsample the received signal by zero padding and low-pass filtering [3]. For example, after 8 times upsampling, the maximum phase measurement error is reduced from $\frac{2\pi}{3}$ to $\frac{2\pi}{3\times8}$ and the corresponding distance measurement error is 0.885 mm.

Suppose that the upsampling factor is M, the sampling rate is F_s, and the time window which containing L peak/valleys at the receiver side contains N sampling points. Since the transmitted signal is a sinusoidal signal with frequency f_0, the phase change $\Delta\phi_t$ can be expressed as:

$$\Delta\phi_t = 2\pi f_0 t = 2\pi f_0 \frac{N}{F_s M} \tag{11}$$

Putting Eq. 9, Eq. 10 and Eq. 11 into Eq. 8, we can estimate the movement distance in the current time window as:

$$\Delta d = \frac{\Delta\phi_r - \Delta\phi_t}{2\pi}\frac{c}{f_0} = c\left(\frac{L}{2f_0} - \frac{N}{F_s M}\right) \tag{12}$$

where c is the speed of sound. From Eq. 12, we can estimate the moving distance of the target based on the number of sampling points N contained in the time window. Note that all other variables are known and N is the only unknown parameter we need to obtain for this distance calculation. Note that if the received sinusoidal wave is compressed, a smaller N value will be obtained and if the received wave is stretched, a larger N value will be obtained.

As shown in Fig. 10, when the parameter settings are as follows: $F_s = 48\ kHz$, $M = 4$, $f_0 = 16000\ Hz$, $L = 4$ and $c = 340\ m/s$, we show a received signal wave and its corresponding estimated transmitted signal wave. We have $N = 20$ sample points in the time window and the calculated movement distance is 7.1 mm, indicating that the received signal wave is compressed. We could also easily infer that $N = 24$ means that there is no relative movement between the transmitter and receiver.

6 TARGET TRACKING

In this section, we first describe how to track an earbud based on the distance measurement scheme presented in Sec. 5. Then, we present the time synchronization issue in 2D/3D tracking and our solution.

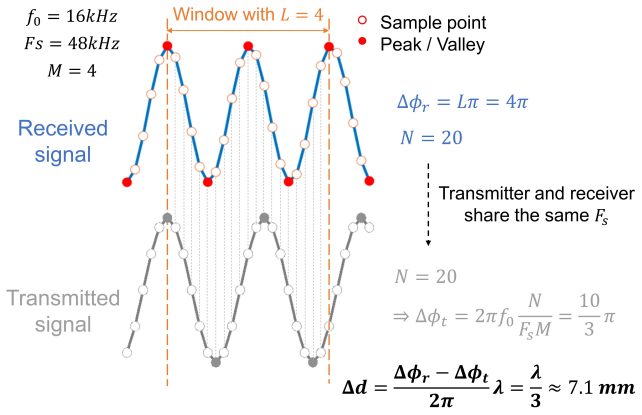


Figure 10: Moving distance calculation

6.1 Tracking earbuds

We can achieve multi-target tracking in 1D/2D/3D using commercial earphones. Without loss of generality, we take 3D tracking of a single earbud as an example to introduce our method.

As shown in Fig. 12, A, B and C are positions of the three microphones, respectively. B is located at the origin (O) of the 3D coordinate system X-Y-Z. A is located on axis OZ and C is located on axis OX and thus $AB \perp BC$. Now a speaker is moved in this 3D coordinate system. Assuming S_{i-1} is the position of the speaker at time $i-1$, we can obtain the new speaker position S_i after a small time interval (e.g, 5 ms) at time i using a distance approximate calculation. Note that for a small time interval, the movement distance $|\overrightarrow{S_{i-1}S_i}|$ is very small. As shown in Fig. 11, when $|\overrightarrow{S_{i-1}S_i}|$ is very small and $|\overrightarrow{S_{i-1}A}|$ is large, angle γ is close to 0 and thus:

$$|\overrightarrow{AS_i}| \approx |\overrightarrow{AS_i}| \cos\gamma = |\overrightarrow{AS_i'}|. \tag{13}$$

We can then approximate $|\overrightarrow{AS_i}|$ as $|\overrightarrow{AS_{i-1}}| - |\overrightarrow{S_{i-1}S_i'}|$.

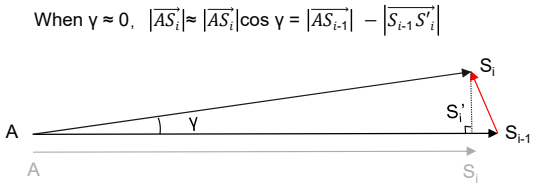


Figure 11: Distance approximate.

We represent the speaker moving distances towards the three microphones from time $i-1$ to time i as Δd_A, Δd_B and Δd_C respectively. Then the distances between the speaker and the three microphones can be expressed as below:

$$\begin{cases} |\overrightarrow{AS_i}| = |\overrightarrow{AS_{i-1}}| - \Delta d_A \\ |\overrightarrow{BS_i}| = |\overrightarrow{BS_{i-1}}| - \Delta d_B \\ |\overrightarrow{CS_i}| = |\overrightarrow{CS_{i-1}}| - \Delta d_C \end{cases} \tag{14}$$

Finally, the position S_i in the 3D coordination system (x_i, y_i, z_i) can be estimated as:

$$\begin{cases} x_i = |\overrightarrow{S_iB}| \cos\alpha \\ z_i = |\overrightarrow{S_iB}| \cos\beta \\ y_i = \sqrt{|\overrightarrow{S_iB}|^2 - x_i^2 - z_i^2} \end{cases} \tag{15}$$

where

$$\begin{cases} \cos\alpha = \frac{|\overrightarrow{BC}|^2 + |\overrightarrow{S_iB}|^2 - |\overrightarrow{S_iC}|^2}{2|\overrightarrow{BC}||\overrightarrow{S_iB}|} \\ \cos\beta = \frac{|\overrightarrow{AB}|^2 + |\overrightarrow{S_iB}|^2 - |\overrightarrow{S_iA}|^2}{2|\overrightarrow{AB}||\overrightarrow{S_iB}|} \end{cases} \tag{16}$$

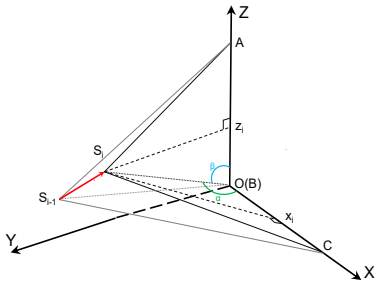


Figure 12: Position update of 3D tracking.

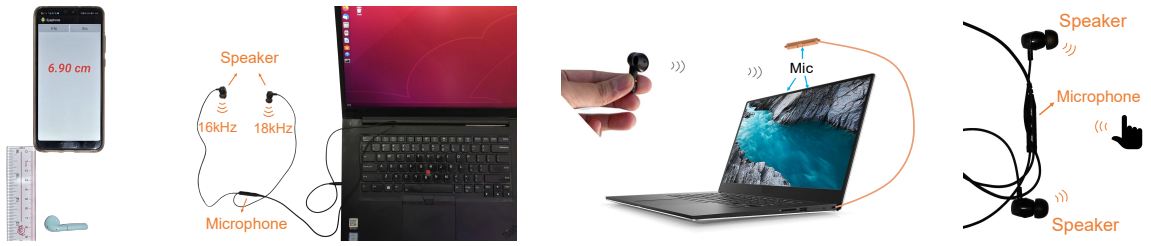
Similarly, we can measure S_0A, S_0B and S_0C, and calculate the initial position S_0 (x_0, y_0, z_0) using Eq. 15.

6.2 Time synchronization

To accurately measure the phase change, we adopt a moving time window that contains a fixed number of peaks/valleys instead of a fixed number of sampling points. Note that due to movement, the signal wave can be compressed or stretched, so a fixed number of peaks/valleys means the time period of the window varies. The windows with varying time periods introduce a time asynchronous problem in our system. The problem is illustrated in Fig. 14. As the window size depends on the relative movement between the transmitter and receiver, thus, the window sizes are different at the three microphones. If we employ time windows of the different sizes for tracking, errors are introduced at those parts windows are not overlapping. To address this issue, when we combine the distance information from multiple microphones, we make sure the windows are fully aligned. If one window (W_{Ai}) is smaller than the other (W_{Bi}), we will append part of the next window $(W_{A_{i+1}})$ to window (W_{Ai}) to fully align it with (W_{Bi}) and the two windows now are of exactly the same size.

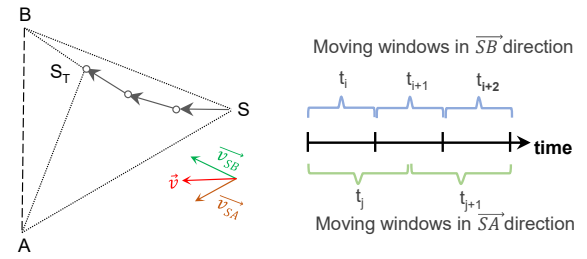
7 FREQUENCY SHIFT COMPENSATION

For wireless earphones, we notice that there exists a frequency shift between the expected signal and the actually transmitted signal, causing a cumulative distance measurement error. We find that the frequency shift is linearly related to the frequency of the generated signal i.e., a signal at frequency $f_0' = (1+\alpha)f_0$ is actually generated



(a) 1D tracking (b) 2D tracking (c) 3D tracking (d) Device-free tracking

Figure 13: Experiment setup



(a) Different observation speed. (b) Misalignment of moving windows for different viewers.

Figure 14: Time asynchronous.

when we try to generate a signal at frequency f_0. Without loss of generality, we assume $\alpha > 0$. Fortunately, the frequency shift coefficient α is stable in each power ON/OFF cycle of the device. Based on Eq. 12, the movement distance is zero if the number of sampling points N in a moving window containing L extreme points satisfies the following equation:

$$N = \frac{F_s \cdot L}{2f_0} \tag{17}$$

When this condition is satisfied, we term the N value as N_{base}. Using N_{base}, Eq. 12 can be rewritten as:

$$d = c\frac{\Delta N}{F_s} = c\frac{N_{base} - N}{F_s} \tag{18}$$

When there is a frequency shift, the actual number of base sampling points we obtain is $N'_{base} = F_s \cdot L/2f'_0$ which is different from the the expected N_{base}. From our experiments, we know that frequency shift is usually less than 2 Hz, and thus the difference between N_{base} and N'_{base} is a small value (usually less than 0.1) rather than an integer. Thus, we cumulate such frequency shift over multiple (P) windows to have one sample shift:

$$N_{base} - N'_{base} = \frac{F_s \cdot L}{2}(\frac{1}{f_0} - \frac{1}{f'_0}) \approx \frac{1}{P}, P \in \mathbb{Z} \tag{19}$$

$$P \times (N_{base} - N'_{base}) \approx 1 \tag{20}$$

For one sample shift, the distance error is $\Delta d = -c/F_s$ and we can compensate this error out.

8 SYSTEM EVALUATION

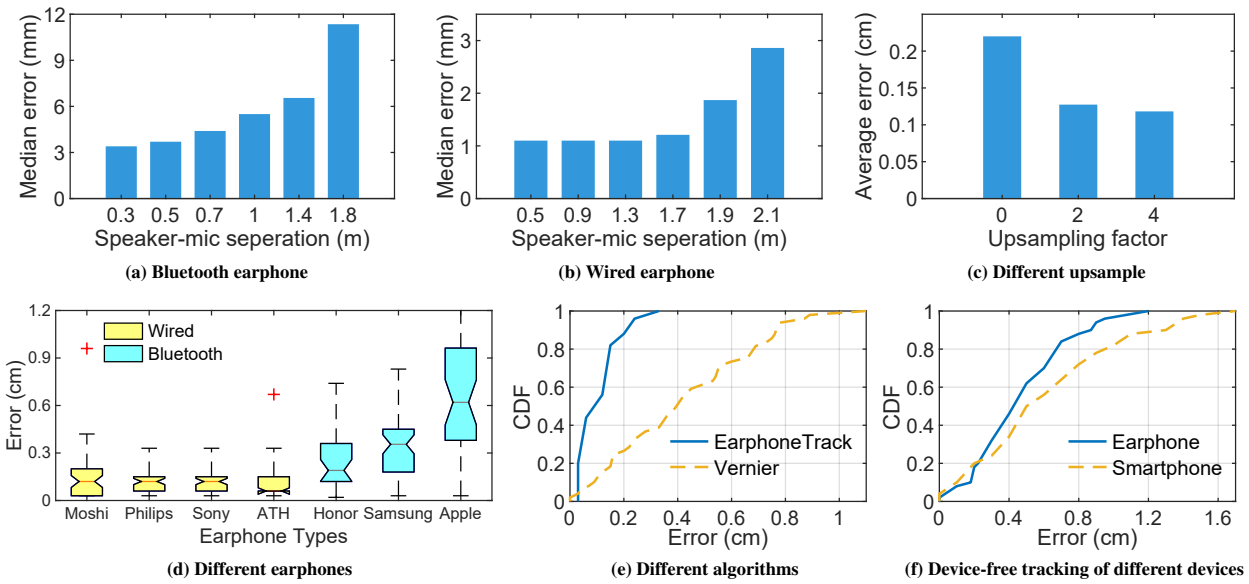
8.1 Implementation

We implement *EarphoneTrack* on both Android smartphones and PC. On the Android platform, we develop an App that emits 16 kHz sine wave signal through a connected wireless earphone and receives the signal through the build-in microphones at a sampling rate of 96 kHz. The App performs signal processing and displays the movement trajectory on the screen in real time. To balance the latency and tracking accuracy, we choose a data segment size containing 360 extreme points and $M = 4$ as the upsampling factor. We conduct wireless earphone tracking on this platform using Huawei P20 and Samsung Galaxy Buds+. The experiment setup is shown in Fig. 13a. Note that the Android platform can support 1D and 2D tracking with both wireless and wired earphones. Due to the space limit, we only present the 1D tracking result using the Andriod platform with a wireless earphone and present 2D/3D tracking results using the earphone-PC combination.

On the Android platform, we also implement the well-known LLAP [54] system to showcase the device-free tracking performance of our system with earphones. The experiment setup is shown in Fig. 13d. We put two speakers and a microphone of a wired earphone (Sony MDR-XB75AP) side by side toward a same direction. The two speakers transmit 16 kHz sinusoidal waves and the microphone captures the signals reflected back from the user's finger to track the finger movement. We also implement our system on PC platform. For 1D tracking, one wired earphone suffices. However, as the microphone and speaker on the same wired earphone are connected by a fixed-length wire, we are not able to separate them far enough to evaluate the effect of microphone-speaker distance on tracking accuracy. Thus, we equip the laptop Lenovo Thinkpad P1 Gen2 [15] with another sound card and connect two wired earphones on the same laptop for tracking. We employ one Sony MDR-XB75AP earphone and one Philips PRO6105BK earphone to show the tracking performance. A speaker of the Sony earphone transmits a 16 kHz sine wave while the microphone on the Philips earphone receives the signal and sends it to PC for processing. For 2D tracking, as shown in Fig. 13b, a single wired earphone Mythro Earbuds is used to transmit and receive signals at the same time, and we need to address the self-interference issue (Sec. 4). The two speakers emit 16 kHz and 18 kHz sine signals respectively and the signals are received at the microphone and sent to the PC for processing. The PC analyzes the audio signal to track the 2D location of the microphone in real time.

In the 2D tracking example, we track the microphone of the earphone. In 3D-tracking, we track the speaker of a wireless earphone. For 3D tracking, three microphones are needed and we employ two built-in microphones on top of the Lenovo Thinkpad P1 Gen2 screen and an external microphone on a wired earphone Philips PRO6105BK, which are shown in Fig. 13c.

We employ an HONOR FlyPods Pro wireless earphone to emit 18 kHz sine wave signals to be tracked in 3D space in real time.



(a) Bluetooth earphone (b) Wired earphone (c) Different upsample

(d) Different earphones (e) Different algorithms (f) Device-free tracking of different devices

Figure 15: 1D Tracking Performance

For 16 kHz and 18 kHz signals, we employ time windows with 320 extreme points and 360 extreme points respectively. The sampling rate and upsampling factor are the same as the Android platform.

8.2 1D Tracking

Overall performance: We first evaluate 1D distance tracking on two different platforms. In this experiment, we vary the initial distance between the microphone and the speaker. The result of the Android platform is shown in Fig. 15a. We move the wireless earphone towards the microphone for a displacement of 20 cm. For each distance, we repeat the experiments 20 times. The results show that the median error is under 4 mm when the speaker-mic separation distance is less than 0.5 m, and increases to 1 cm when the separation is 1.8 m. The 1D tracking results of PC with a wired earphone is shown in Fig. 15b. The cumulative error is well below 2 mm when the separation is less than 2 m. From these results, we can see that the second platform (PC & Wired earphone) achieves higher accuracy than the first one (Smartphone & wireless earphone). One reason is that it is difficult to obtain the very precise position of the microphone inside the smartphone and this factor brings in some errors. Also for wireless earphone, even the frequency shift is compensated, there are still residual errors.

Impact of upsampling factor M: We evaluate the impact of the upsampling factor and the results are shown in Fig. 15c. For 96 kHz sampling rate with no upsampling, the average cumulative error for tracking a 20 cm movement is 2.2 mm. The average error for 2 and 4 times upsampling is decreased to 1.3 mm and 1.1 mm respectively. Compared with no upsampling, an upsampling factor of 4 is able to reduce the error by 50%.

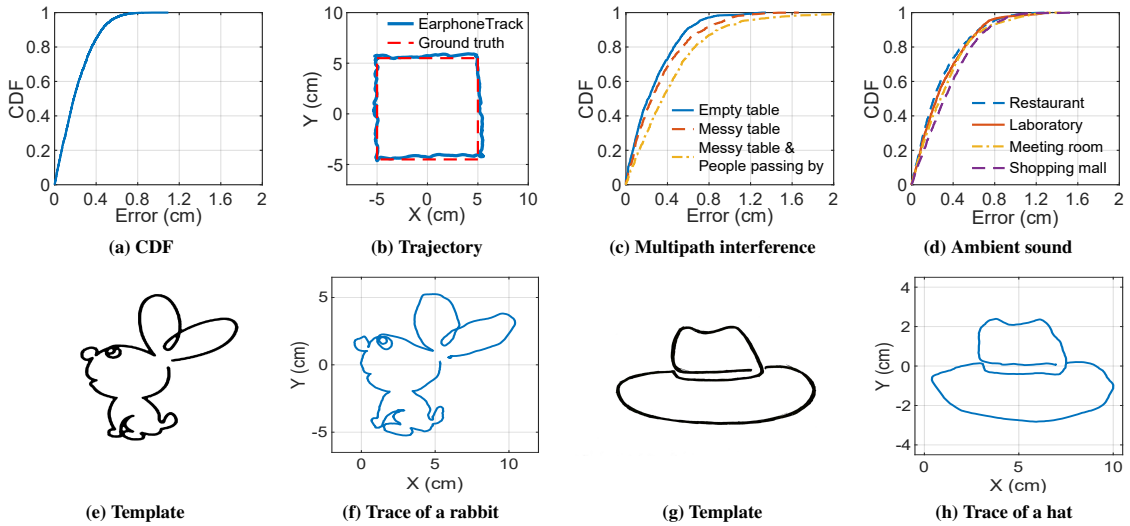
Impact of earphone diversity: We further evaluate our system using different earphones, including four wired earphones [11, 16, 17, 19] and three wireless earphone [12, 21, 23]. We move each earphone for a distance of 20 cm and repeat each experiment 50 times. The results are shown in Fig. 15d. For all four wired earphones, the 75-percentile errors are below 2 mm. The 75-percentile errors for Honor Flypods Pro and Samsung Galaxy Buds+ are below 4.5 mm. For Honor Flypod Pro, its median error is 1.9 mm, which is the best among the 3 wireless earphones. Surprisingly, the Apple Airpods 2 performs not so well due to its weak capability to transmit high frequency sound (at 16 kHz, the signal strength from Airpod 2 is just 1/3 of that from Samsung) and a larger frequency shift (Sec. 2.4).

Comparison with the state-of-the-arts: We do not compare the performance of our system with the chirp-based approaches because the large frequency band is not available with earphones and it is thus not fair to compare the chirp-based systems with our single-frequency-based system. We compare the achieved performance with Vernier [63], another approach based on single frequency signal. We re-implement Vernier and compare its tracking accuracy with our system using the same devices and exactly the same setup. The Cumulative Distribution Function (CDF) plot of the tracking error for a movement of 20 cm is shown in Fig. 15e. The median error of *EarphoneTrack* is 1.1 mm and 90-percentile error is 2 mm, while that of Vernier are 4 mm and 7.7 mm respectively.

Device-free tracking: We compare the performance of device-free tracking between earphones and smartphones. We move a plastic card 10 cm away from the microphone for a displacement of 10 cm and repeat the experiment 50 times. The CDF plots of tracking errors are shown in Fig. 15f. The median error is 4.3 mm and 5 mm for earphone and smartphone, respectively. The achieved accuracies are comparable. However, we note that due to weaker signals, the earphone can hardly track an object further than 30 cm away while the smartphone can support a tracking range up to 50 cm.

8.3 2D Tracking

Overall performance: We now evaluate the tracking accuracy in 2D case. In this experiment, we ask five volunteers to draw a 10 cm $\times$ 10 cm square ten times by holding the microphone of the wired earphone. The start position of microphone is 30 cm away from the



(a) CDF **(b) Trajectory** **(c) Multipath interference** **(d) Ambient sound**

(e) Template **(f) Trace of a rabbit** **(g) Template** **(h) Trace of a hat**

Figure 16: 2D Tracking Performance

speaker. Fig. 16b shows the trajectory of one drawing example and Fig. 16a shows the CDF of the tracking error for all the volunteers. The median error is 1.9 mm and the 90-th percentile error is 4.6 mm.

Impact of environment diversity: We evaluate the robustness of our system in different environments for 2D tracking. The first one is an empty table with no other objects. For the second environment, we put books and bottles on the table and these items are 10 - 15 cm away from the speaker and microphone. For the third environment, we further ask two volunteers to walk around near the table (about 50 *cm* from the device) continuously to create multipath reflections. The three different environments have low, medium and rich multipath, respectively. We ask five volunteers to draw a 10 cm × 10 cm square five times in each environment. The results are shown in Fig. 16c. The median error are 2.1 mm, 2.5 mm and 3.5 mm respectively for the three environments. These results show that with more multipath, the tracking accuracy does decrease. However, as the reflection signals are much weaker than the LoS signal, the performance is not affected much.

Impact of ambient sound noise: To evaluate the robustness of the proposed system against ambient sound noise, We evaluate the 2D tracking performance in four different environments with different levels of sound noise, including a restaurant, a laboratory, a meeting room and a shopping mall. Fig. 16d shows the tracking error in these environments. We do not observe a significant difference among the four environments. We believe this is because the frequencies of the ambient noise are much lower than the frequency adopted for tracking and thus the ambient noise has little effect on the performance of *EarphoneTrack*.

Fine-grained drawing: Fig. 16f and 16h show two more complex 2D drawing samples using *EarphoneTrack*. We can see that the drawn rabbit and hat match the ground-truths very well. These results show that the accuracy achieved by *EarphoneTrack* in real time is fine enough to enable a lot of HCI applications. Compared with other acoustic motion tracking schemes [33, 50, 60, 63], our system based on earphones can achieve similar accuracy and exhibits unique advantages on flexibility.

Impact of user diversity: We further evaluate the robustness of our system across different users. We ask six users to draw a 10 cm × 10 cm square five times. The median errors for different users are in the range of 1.5 mm - 4 mm. We believe the performance difference can be due to different drawing speeds among users. Another reason is that some users do not follow the square template precisely during the drawing process.

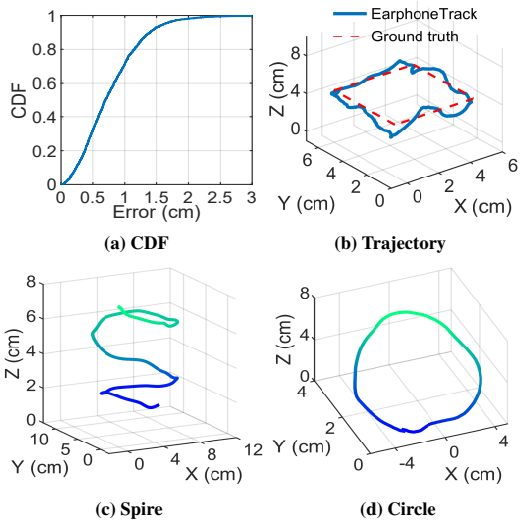
8.4 3D Tracking

For 3D tracking, we measure the tracking accuracy by drawing a square in the 3D space. The ground-truth is a 5 cm × 5 cm square at the height of 4.5 cm. The coordinate of the three microphones in Fig. 17b are $(-30, 0, 5)$, $(-30, 0, -5)$, $(-30, 8, -5)$ cm respectively. We ask a user to draw a square 100 times. Fig. 17b shows the trajectory of one drawing example and Fig. 17a shows the CDF plot of the tracking error. The 50-th percentile error is 6.9 mm, which is slightly larger than 2D case because of a higher degree of freedom. Fig. 17c and 17d show the drawing examples of a spire and a circle in 3D space. These results demonstrate that *EarphoneTrack* could enable accurate 3D motion tracking.

9 DISCUSSIONS

Microphones Bluetooth earphones: Surprisingly, most commercial wireless earphones are not able to receive sound signals of a frequency above 10 kHz. So wireless earphones in our tracking modality only act as the signal generator and additional microphones are required for signal reception.

Sensing/tracking range: Limited by the signal strength, *EarphoneTrack* can support accurate tracking when the distance between the speaker and microphone is less than 1.5 m. When the distance is larger than 1.5 m, the tracking accuracy degrades significantly. In addition, for 2D tracking using one wired earphone, the distance is also constrained by the length of the cable between the speaker and microphone. Therefore, *EarphoneTrack* is more suitable for motion tracking in a small area, such as drawing on a table.



(a) CDF (b) Trajectory

(c) Spire (d) Circle

Figure 17: 3D tracking accuracy

10 RELATED WORK

We discuss the literature most related to our work below:

Acoustic-based tracking: Recently, quite a few acoustic-based tracking schemes [33, 36, 38, 41, 45, 50, 54, 57, 60, 61, 63, 64] have been proposed. BeepBeep [38] accurately estimates the propagation time of sound signal after removing the indeterministic latency caused by hardware and software. SwordFight [64] optimizes BeepBeep [38] and makes it applicable to real-time mobile games. These systems can achieve a *cm*-level accuracy, which is still coarse. CAT [33] and MilliSonic [50] employ Frequency Modulated Continuous Waveform (FMCW) signals to address multipath interference and estimate the propagation delay. These systems are able to achieve real-time tracking at a higher accuracy. However, the FMCW-based schemes require a relatively large bandwidth which is not available at earphones. Based on the Doppler shift, AAMouse [60] can track a smartphone at an accuracy of 1.4 cm. However, AAMouse [60] can only work reliably for a short interval (*i.e.*, less than 5 seconds) due to the accumulation of errors. The closest literature to our work is Venier [63], which achieves real-time motion tracking with an error less than 4 mm. PAMT [31] defines Multipath Effect Ration (MER) as a metric to evaluate the impact of multipath fading on the signal at different frequencies, and selects 'clean' signals to estimate the moving distance. The measurement errors of PAMT are 2 mm and 4 mm in 1-D and 2-D scenarios. To achieve this performance, PAMT requires extra hardware and the computational cost is high. FingerIO [36], LLAP [54], Strata [61] and Vskin [45] are latest device-free motion tracking systems focusing on smartphone-based tracking.

Non-acoustic tracking: Besides acoustic signals, a lot of Radio Frequency (RF) signals are employed for sensing. Vision-based and the IMU-based systems are also popular. RF signals have been widely used for localization and tracking [1, 46, 48, 51, 53, 56, 58]. ArrayTrack [56] achieves a 30 cm indoor location accuracy. WiTrack [1] can track the target through a wall using the FMCW signal. Widraw [46] enables hand-free drawing in the air. Tagoram [58] proposes the Differential Augmented Hologram (DAH) scheme and can track the target at a *mm*-level accuracy. MilliBack [55] develops a backscatter-based handwriting tracking system in 2D using customized hardware. In addition to RF-based schemes, vision-based schemes are also proposed to track motion using cameras or light sensors [13, 28, 62]. OKuli achieves a location accuracy of 7 mm using one LED emitter and two photodetectors. The Sony PlayStation VR [28] system employs a separate camera to tracking LED markers on the headset and controllers. Despite being accurate enough for motion tracking, these systems are sensitive to lighting conditions and the performance degrades sharply in the presence of strong ambient light. Inertial Measurement Unit (IMU) based schemes [9, 25, 44] are only applicable to coarse-grained tracking because the IMU measurements are very coarse due to gravity pollution, magnetic interference and inherent sensor noise [44]. Acoustic-based schemes are more appropriate for fine-grained motion tracking.

Sensing based on earable devices: Earable device with sensors can serve as a physiological parameter monitor. In a recent work [37], a smart earable device integrated with an infrared sensor is used to detect the body temperature. Salustek [18] can detect vital signs with a conventional earphones. Earable RCC [47] develops a chewing-counting measurement device that provides real-time visualization of chewing movements and the number of chews. Another recent work [35] can recognize human activities such as nodding, shaking, walking, stepping up, speaking and so on, using earable devices with a 6-axis inertial measurement unit and a microphone. Earable devices are also used for human-computer interaction. Headphone Taps [32] detects tapping on the earphone shell by using the speakers as sensors. [43] designs an earable device with biosignal sensors and uses it as a controller for applications such as automatic music select, tactile communication, and automatic metadata annotation.

11 CONCLUSIONS

In this paper, we present *EarphoneTrack*, the first earphone-based motion tracking system, which can track users' motions in real time at a *mm*-level accuracy. We believe earphone tracking is a promising new acoustic tracking modality which has a great potential to enable a large range of applications. We propose solutions to address several unique challenges associated with earphone motion tracking and implement *EarphoneTrack* on commodity hardware. Comprehensive experiments demonstrate the feasibility and great flexibility of employing earphones for fine-grained motion tracking.

ACKNOWLEDGMENTS

Xiang-Yang Li and Panlong Yang are the co-corresponding authors. This research is partially supported by National Key R&D Program of China 2018YFB0803400, China National Funds for Distinguished Young Scientists with No.61625205, China National Natural Science Foundation with No.61751211, 61520106007, Key Research Program of Frontier Sciences, CAS. No.QYZDY-SSW-JSC002, NSFC with No.62072424, 61772546, 61632010, PCL Future Greater-Bay Area Network Facilities for Large-scale Experiments and Applications(LZC0019), the University Synergy Innovation Program of Anhui Province with No.GXXT-2019-024, Key Research Program of Frontier Sciences, CAS. No. ZDBS-LY-JSC001.

REFERENCES

[1] Fadel Adib, Zachary Kabelac, Dina Katabi, and Rob Miller. 2014. WiTrack: motion tracking via radio reflections off the body. In *Proc. of NSDI*.

[2] Sandip Agrawal, Ionut Constandache, Shravan Gaonkar, Romit Roy Choudhury, Kevin Caves, and Frank DeRuyter. 2011. Using mobile phones to write in air. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*. 15–28.

[3] John R. Buck Alan V. Oppenheim, Ronald W. Schafer. 1989. *Discrete-time signal processing*. Vol. volume 2. Prentice-Hall, Inc.

[4] Inc Amazon.com. 2019. Amazon Alexa. https://developer.amazon.com/en-US/alexa.

[5] Inc Apple. 2019. AirPods. https://www.apple.com/shop/product/MRXJ2AM/A/airpods-with-wireless-charging-case.

[6] Inc Arduino.com. 2019. Arduino. https://www.arduino.cc/.

[7] Inc Audio-Technica. 2019. ATH-CKL220iS. https://audio-technica.com.au/products/ath-ckl220is/.

[8] Inc Bela.com. 2019. Bela. https://www.blta.com.

[9] Kongyang Chen and Guang Tan. 2018. BikeGPS: Accurate Localization of Shared Bikes in Street Canyons via Low-Level GPS Cooperation. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 150–162.

[10] Zicheng Chi, Yao Yao, Tiantian Xie, Xin Liu, Zhichuan Huang, Wei Wang, and Ting Zhu. 2018. EAR: Exploiting uncontrollable ambient RF signals in heterogeneous networks for gesture recognition. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. 237–249.

[11] Audio-Technica Corporation. 2019. ATH-CK350IS. http://sea.audio-technica.com/happenings/2018-05/audio-technica-unveils-ath-ck350is-new-in-ear-headphones-for-smartphones.

[12] HUAWEI Corporation. 2019. Flypods Lite. https://www.hihonor.com/global/products/accessories/honor-flypods-lite/.

[13] HTC Corporation. 2019. HTC VIVE. https://www.vive.com/us/product/vive-virtual-reality-system/.

[14] HUAWEI Corporation. 2019. HUAWEI P20. https://consumer.huawei.com/en/support/phones/p20/.

[15] Lenovo Corporation. 2019. ThinkPad P1 Gen2. https://www.lenovo.com/us/en/laptops/thinkpad/thinkpad-p/P1-Gen-2/p/22WS2WPP102.

[16] Moshi Corporation. 2019. Mythro Earbuds. https://www.moshi.com/en/product/audio-earbuds-mythro-mic/gray.

[17] Philips Corporation. 2019. Philips PRO6105BK. https://www.usa.philips.com/c-p/PRO6105BK_00/6000-series-in-ear-headphones-with-mic.

[18] SaLusTex Corporation. 2019. Exhibit at Healthcare Next Generation. https://salustek.com/en/25/.

[19] Sony Corporation. 2019. Sony MDR-XB75AP. https://www.sony.com.my/electronics/in-ear-headphones/mdr-xb75ap.

[20] Sony Corporation. 2019. WF-1000XM3. https://helpguide.sony.net/mdr/wf1000xm3/v1/en/contents/TP0002289865.html.

[21] Samsung Corporation. 2020. Samsung Galaxy Buds+. https://www.samsung.com/us/mobile/audio/galaxy-buds-plus/.

[22] Inc Google.com. 2019. Google Home. https://store.google.com/gb/product/google_home.

[23] Apple Inc. 2019. AirPods Pro. https://support.apple.com/kb/SP811?viewlocale=en_US&locale=en_US.

[24] AVTech Media Americas Inc. 2019. Headphone measures. https://www.innerfidelity.com/headphone-measurements.

[25] Google Inc. 2019. Google Daydream. https://arvr.google.com/daydream/.

[26] Quebec Inc. 2019. Headphones Frequency Response. https://www.thephonograph.net/headphones-frequency-response/.

[27] Quebec Inc. 2019. Raw Frequency Response. https://www.rtings.com/headphones/tests/sound-quality/raw-frequency-response.

[28] Sony Interactive Entertainment Inc. 2019. PlayStation VR. https://www.playstation.com/.

[29] Pravein Govindan Kannan, Seshadri Padmanabha Venkatagiri, Mun Choon Chan, Akhihebbal L Ananda, and Li-Shiuan Peh. 2012. Low cost crowd counting using audio tones. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. 155–168.

[30] Samuel J. Ling, Jeff Sanny, and William Moebs. 2016. University Physics - Volume 2 (OpenStax). (2016).

[31] Yang Liu, Wuxiong Zhang, Yang Yang, Weidong Fang, Fei Qin, and Xuewu Dai. 2019. PAMT: Phase-based Acoustic Motion Tracking in Multipath Fading Environments. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2386–2394.

[32] Hiroyuki Manabe and Masaaki Fukumoto. 2012. Headphone taps: A simple technique to add input function to regular headphones. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services companion*.

[33] Wenguang Mao, Jian He, and Lili Qiu. 2016. CAT: high-precision acoustic motion tracking. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 69–81.

[34] Robert A. Meyers. 1987. *Encyclopedia of physical science and technology*.

[35] Chulhong Min, Akhil Mathur, and Fahim Kawsar. 2018. Exploring audio and kinetic sensing on earable devices. In *Proceedings of the 4th ACM Workshop on Wearable Systems and Applications*. ACM, 5–10.

[36] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. (2016).

[37] Hiroki Ota, Minghan Chao, Yuji Gao, Eric Wu, Li-Chia Tai, Kevin Chen, Yasutomo Matsuoka, Kosuke Iwai, Hossain M Fahad, Wei Gao, et al. 2017. 3d printed "earable" smart devices for real-time detection of core body temperature. *ACS sensors* 2, 7 (2017), 990–997.

[38] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. 2007. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *Proceedings of the 5th international conference on Embedded networked sensor systems*. ACM, 1–14.

[39] Alexander D. Poularikas. 1998. *The handbook of formulas and tables for signal processing*.

[40] Swadhin Pradhan, Eugene Chai, Karthikeyan Sundaresan, Lili Qiu, Mohammad A Khojastepour, and Sampath Rangarajan. 2017. Rio: A pervasive rfid-based touch gesture interface. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. 261–274.

[41] Nissanka B Priyantha, Anit Chakraborty, and Hari Balakrishnan. 2000. The cricket location-support system. In *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM, 32–43.

[42] Wenjie Ruan, Quan Z Sheng, Lei Yang, Tao Gu, Peipei Xu, and Longfei Shangguan. 2016. AudioGest: enabling fine-grained hand gesture detection by decoding echo signal. In *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*. 474–485.

[43] Akane Sano, Takashi Tomita, and Haruo Oba. 2010. Applications using earphone with biosignal sensors. In *Human Interface Society Meeting*, Vol. 12. 1–6.

[44] Sheng Shen, Mahanth Gowda, and Romit Roy Choudhury. 2018. Closing the gaps in inertial motion tracking. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 429–444.

[45] Ke Sun, Ting Zhao, Wei Wang, and Lei Xie. 2018. 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*. ACM, 591–605.

[46] Li Sun, Souvik Sen, Dimitrios Koutsonikolas, and Kyu-Han Kim. 2015. Widraw: Enabling hands-free drawing in the air on commodity wifi devices. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 77–89.

[47] Kazuhiro Taniguchi, Hisashi Kondo, Toshiya Tanaka, and Atsushi Nishikawa. 2018. Earable RCC: development of an earphone-type reliable chewing-count measurement device. *Journal of healthcare engineering* 2018 (2018).

[48] Deepak Vasisht, Swarun Kumar, and Dina Katabi. 2016. Decimeter-level localization with a single WiFi access point. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*. 165–178.

[49] Aditya Virmani and Muhammad Shahzad. 2017. Position and orientation agnostic gesture recognition using wifi. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. 252–264.

[50] Anran Wang and Shyamnath Gollakota. 2019. MilliSonic: Pushing the Limits of Acoustic Motion Tracking. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 18.

[51] Chuyu Wang, Jian Liu, Yingying Chen, Hongbo Liu, Lei Xie, Wei Wang, Bingbing He, and Sanglu Lu. 2018. Multi-touch in the air: Device-free finger tracking and gesture recognition via COTS RFID. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 1691–1699.

[52] Hao Wang, Daqing Zhang, Junyi Ma, Yasha Wang, Yuxiang Wang, Dan Wu, Tao Gu, and Bing Xie. 2016. Human respiration detection with commodity wifi devices: do user location and body orientation matter?. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. 25–36.

[53] Jue Wang, Deepak Vasisht, and Dina Katabi. 2015. RF-IDraw: virtual touch screen in the air using RF signals. *ACM SIGCOMM Computer Communication Review* 44, 4 (2015), 235–246.

[54] Wang Wei, Alex X. Liu, and Sun Ke. 2016. Device-free gesture tracking using acoustic signals. In *International Conference on Mobile Computing & Networking*.

[55] Ning Xiao, Panlong Yang, Xiang-Yang Li, Yanyong Zhang, Yubo Yan, and Hao Zhou. 2019. 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 (2019), 112.

[56] Jie Xiong and Kyle Jamieson. 2013. Arraytrack: A fine-grained indoor location system. In *Presented as part of the 10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13)*. 71–84.

[57] Jie Yang, Simon Sidhom, Gayathri Chandrasekaran, Tam Vu, Hongbo Liu, Nicolae Cecan, Yingying Chen, Marco Gruteser, and Richard P Martin. 2011. Detecting driver phone use leveraging car speakers. In *Proceedings of the 17th annual international conference on Mobile computing and networking*. ACM, 97–108.

[58] Lei Yang, Yekui Chen, Xiang-Yang Li, Chaowei Xiao, Mo Li, and Yunhao Liu. 2014. 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. ACM, 237–248.

[59] Shichao Yue, Hao He, Hao Wang, Hariharan Rahul, and Dina Katabi. 2018. Extracting multi-person respiration from entangled RF signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 2 (2018), 1–22.

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

[61] Sangki Yun, Yi-Chao Chen, Huihuang Zheng, Lili Qiu, and Wenguang Mao. 2017. Strata: Fine-grained acoustic-based device-free tracking. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 15–28.

[62] Chi Zhang, Josh Tabor, Jialiang Zhang, and Xinyu Zhang. 2015. Extending mobile interaction through near-field visible light sensing. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 345–357.

[63] Yunting Zhang, Jiliang Wang, Weiyi Wang, Zhao Wang, and Yunhao Liu. 2018. Vernier: Accurate and Fast Acoustic Motion Tracking Using Mobile Devices. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 1709–1717.

[64] Zengbin Zhang, David Chu, Xiaomeng Chen, and Thomas Moscibroda. 2012. Swordfight: Enabling a new class of phone-to-phone action games on commodity phones. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 1–14.