Device-Free Gesture Tracking Using Acoustic Signals

Wei Wang[†] Alex X. Liu[†‡] Ke Sun[†]
[†]State Key Laboratory for Novel Software Technology, Nanjing University, China
[‡]Dept. of Computer Science and Engineering, Michigan State University, USA
ww@nju.edu.cn, alexliu@cse.msu.edu, samsonsunke@gmail.com

ABSTRACT

Device-free gesture tracking is an enabling HCI mechanism for small wearable devices because fingers are too big to control the GUI elements on such small screens, and it is also an important HCI mechanism for medium-to-large size mobile devices because it allows users to provide input without blocking screen view. In this paper, we propose LLAP, a device-free gesture tracking scheme that can be deployed on existing mobile devices as software, without any hardware modification. We use speakers and microphones that already exist on most mobile devices to perform device-free tracking of a hand/finger. The key idea is to use acoustic phase to get fine-grained movement direction and movement distance measurements. LLAP first extracts the sound signal reflected by the moving hand/finger after removing the background sound signals that are relatively consistent over time. LLAP then measures the phase changes of the sound signals caused by hand/finger movements and then converts the phase changes into the distance of the movement. We implemented and evaluated LLAP using commercial-off-the-shelf mobile phones. For 1-D hand movement and 2-D drawing in the air, LLAP has a tracking accuracy of 3.5 mm and 4.6 mm, respectively. Using gesture traces tracked by LLAP, we can recognize the characters and short words drawn in the air with an accuracy of 92.3% and 91.2%, respectively.

CCS Concepts

•**Human-centered computing** → **Gestural input;**

Keywords

Gesture Tracking; Ultrasound; Device-free

1. INTRODUCTION

1.1 Motivation

Gestures are natural and user-friendly Human Computer Interaction (HCI) mechanisms for users to control their devices. Gesture tracking allows devices to get fine-grained user input by quantitatively measuring the movement of their hands/fingers in the air. Device-free gesture tracking means that user hands/fingers are not attached with any device. Imagine that if a smart watch has the device-free gesture tracking capability, then the user can adjust time in a touch-less manner as shown in Figure 1, where the clock hand follows the movement of the finger. Device-free gesture tracking is an enabling HCI mechanism for small wearable devices (such as smart watches) because fingers are too big to control the GUI elements on such small screens. In contrast, device-free gesture tracking allows users to provide input by performing gestures *near* a device rather than *on* a device. Device-free gesture tracking is also an important HCI mechanism for medium-to-large size mobile devices (such as smartphones and tablets) complementing touch screens because it allows users to provide inputs without blocking screen view, which gives user better visual experience. Furthermore, device-free gesture tracking can work in scenarios where touch screens cannot, *e.g.*, when users wear gloves or when the device is in the pocket.

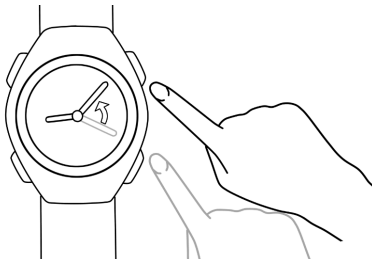


Figure 1: Device-free gesture tracking

Practical device-free gesture tracking systems need to satisfy three requirements. First, such systems need to have *high accuracy* so that they can capture delicate movements of a hand/finger. Due to the small operational space around the mobile device, *e.g.*, within tens of centimeters (cm) to the device, we need millimeter (mm) level tracking accuracy to fully exploit the control capability of human hands. Second, such systems need to have *low latency* (*i.e.*, respond quickly), within tens of milliseconds, to hand/finger movement without user feeling lagging responsiveness. Third, they need to have *low computational cost* so that they can be implemented on resource constrained mobile devices.

1.2 Limitations of Prior Art

Most existing device-free gesture tracking solutions use customized hardware [1–4]. Based on the fact that wireless signal changes as a hand/finger moves, Google made a customized chip in their Soli system that uses 60 GHz wireless signal with mm-level wavelength to track small movement of a hand/finger [1], and Teng *et al.* made customized directional 60 GHz transceivers

MobiCom'16, October 03-07, 2016, New York City, NY, USA

DOI: http://dx.doi.org/10.1145/2973750.2973764

in their mTrack system to track the movement of a pen or a finger using steerable directional beams [2]. Based on the fact that light reflection strength changes as a hand/finger moves, Zhang *et al.* made customized LED/light sensors in their Okuli system to use visible light to track hand/finger movement [3]. Based on vision processing algorithms, Leap Motion made customized infrared cameras to track hand/finger movements [4]. Recently, Nandakumar *et al.* explored the feasibility of using commercial mobile devices to track fingers/hands within a short distance. They proposed fingerIO, which uses OFDM modulated sound to locate the fingers with accuracy of 8 mm [5].

1.3 Proposed Approach

In this paper, we propose a device-free gesture tracking scheme, called Low-Latency Acoustic Phase (LLAP), that can be deployed on existing mobile devices as a software (such as an APP) without any hardware modification. We use speakers and microphones that already exist on most mobile devices to perform device-free tracking of a hand/finger. Commercial-Off-The-Shelf (COTS) mobile devices can emit and record sound waves with frequency higher than 17 kHz, which are inaudible to most people [6]. The wavelength of sound waves in this frequency range is less than 2 *cm*. Therefore, a small movement of a few millimeters will significantly change the phase of the received sound wave. Our key idea is to use the acoustic phase to get fine-grained movement direction and movement distance measurements. LLAP first extracts the sound signal reflected by the moving hand/finger after removing the background sound signals that are relatively consistent over time. Second, LLAP measures the phase changes of the sound signals caused by hand/finger movements and then converts the phase changes into the distance of the movement. LLAP achieves a tracking accuracy of 3.5 mm and a latency of 15 ms on COTS mobile phones with limited computing power. For mobile devices with two or more microphones, LLAP is capable of 2-D gesture tracking that allows users to draw in the air with their hands/fingers.

1.4 Technical Challenges and Solutions

The first challenge is to achieve mm-level accuracy for the measurement of hand/finger movement distance. Existing sound based ranging systems either use the Time-Of-Arrival/Time-Difference-Of-Arrival (TOA/TDOA) measurements [7, 8] or the Doppler shift measurements [9, 10]. Traditional TOA/TDOA based systems require the device to emit bursty sound signals, such as pulses or chirps, which are often audible to humans as these signals change abruptly [7, 8]. Furthermore, their distance measurement accuracy is often in the scale of cm, except for the recent OFDM phase based approach [5]. Doppler shift based device-free systems do not have tracking capability and can only recognize predefined gestures because Doppler shift can only provide the coarse-grained measurement of the speed or direction of hand/finger movements due to the limited frequency measurement precision [9, 11, 12]. In contrast, to achieve mm-level hand/finger tracking accuracy, we leverage the fact that the sound reflected by a human hand is *coherent* to the sound emitted by the mobile device. Two signals are coherent if they have a constant phase difference and the same frequency. This coherency allows us to use a coherent detector to convert the received sound signal into a complex-valued baseband signal. Our approach is to first measure the phase change of the reflected signal, rather than using the noise-prone integration of the Doppler shift as AAMouse [13] did, and then convert the phase change to the movement distance of a hand/finger. Compared with traditional TOA/TDOA, our approach has two advantages: (1) human inaudibility, and (2) mm-level tracking accuracy. Compared with Doppler shift, our approach has three advantages: (1) tracking capability, (2) low latency, and (3) ability to track slow or small movements of a hand/finger. We have lower latency than Doppler shift based systems because Doppler shift requires Fast Fourier Transform (FFT), which needs to accumulate at least 2048 samples (translated to 42.7 ms) to process, whereas we only need to accumulate 16 samples (translated to 0.3 ms). In other words, Doppler shift based systems only respond to hand/finger movement every 42.7 ms whereas our LLAP system can respond to hand/finger movement every 0.3 ms. Note that in practice, we may need to accumulate more samples due to the hardware limitations of mobile devices, such as 512 samples (translated to 10.7 ms) on smartphones. We can deal with *slow* hand/finger movement because LLAP can precisely measure the accumulated slow phase changes over time. We can deal with *small* hand/finger movement because LLAP can precisely measure small phase changes that is less than a full phase cycle. In contrast, Doppler-based approaches cannot detect slow or small movements due to their limited frequency resolution, as we show in Section 3.

The second challenge is to achieve two dimensional gesture tracking. Although LLAP can precisely measure the relative movement distance of a hand, it cannot directly measure the absolute distance between the hand and the speaker/microphones, and therefore it is hard to determine the initial hand location that is essential for two dimensional tracking. To address this challenge, we use multiple Continuous Waves (CW) with linearly spaced frequencies to measure the path length. We observe that sound waves with different frequencies have different wavelengths, which leads to different phase shifts even if they travel through the same path. To determine the path length of the reflected sound wave, we first isolate the phase changes caused by hand/finger movement and then apply Inverse Discrete Fourier Transform (IDFT) on the phases of different sound frequencies to get the TOA of the path. By identifying the TOA that has the strongest energy in the IDFT result, we can determine the path length for the sound reflected by the moving hand/finger. Thus, our approach can serve as a coarse-grained initial position estimation. Combining the fine-grained relative distance measurement and the coarse-grained initial position estimation, we can achieve a relatively accurate 2-D hand/finger tracking.

1.5 Summary of Experimental Results

We implemented and evaluated LLAP using commercial mobile phones without any hardware modification. Under normal indoor noise level, for 1-D hand movement and 2-D drawing in the air, LLAP has a tracking accuracy of 3.5 mm and 4.57 mm, respectively. Under loud indoor noise level such as playing music, for 1-D hand movement and 2-D drawing in the air, LLAP has a tracking accuracy of 5.81 mm and 4.89 mm, respectively. Experimental results also show that LLAP can detect small hand/finger movements. For example, for a small single-finger movement of 5 mm, LLAP has a detection accuracy of 94% within a distance of 30 cm. Using gesture traces tracked by LLAP, we can recognize the characters and short words drawn in the air with an accuracy of 92.3% and 91.2%, respectively.

2. RELATED WORK

Sound Based Localization and Tracking: TOA and TDOA ranging systems using sound waves has a good ranging accuracy of a few centimeters because of the slower propagation speed compared to radio waves [7, 8, 14–16]. However, such systems often either require specially designed ultrasound transceivers [14] or emit audible probing sounds, such as short bursty sound pulses or chirps [7, 8, 15]. Furthermore, most existing sound based tracking systems are not device-free as they can only track a device that

transmits or receives sound signals [7, 8, 10, 13–15, 17]. For example, AAMouse measures the Doppler shifts of the sound waves transmitted by a smart phone to track the phone itself with an accuracy of 1.4 cm [13]. In comparison, our approach is device-free as we use the sound signals reflected by a hand/finger. The problems that we face are more challenging because the signal reflected by the object has much weaker energy compared to the signal travelled through the Line-Of-Sight (LOS) path.

Sound Based Device-Free Gesture Recognition: Most sound based device-free gesture recognition systems use the Doppler effect of the sound reflected by hands [9, 11, 12]. Such systems do not have tracking capability and can only recognize predefined gestures because Doppler shift can only provide the coarse-grained measurement of the speed or direction of hand/finger movements due to the limited frequency measurement precision [9,11,12]. Another system, ApenaApp, uses chirp signals to detect the changes in reflected sound that is caused by human breaths [18]. ApenaApp applies FFT over the sound signals of a long duration to achieve better distance resolution at the cost of reducing the time resolution. Thus, ApenaApp's approach can only be used for long term monitoring for periodical movements (such as human breaths) that have frequency lower than 1 Hz. There are keystroke recognition systems that use the sound emitted by gestures, such as typing on a keyboard or tapping on a table, to recognize keystrokes [19–21] or handwriting [22]. Compared with such systems, we use inaudible, rather than audible, sound reflected by hands/fingers.

In recent pioneer work parallel with us, Nandakumar *et al.* proposed an OFDM based finger tracking system, called fingerIO [5]. FingerIO achieves a finger location accuracy of 8 mm and also allows 2-D drawing in the air using COTS mobile devices. The key difference between LLAP and fingerIO is that LLAP uses CW signals rather than OFDM pulses. The phase measured by CW signals is less noisy due to the narrower bandwidth compared to OFDM pulses. This allows LLAP to achieve better tracking accuracy. Furthermore, the complex valued baseband signal extracted by LLAP can potentially give more information about hand/finger movements than the TOA measurements from fingerIO. However, the CW signal approach used by LLAP is more susceptible to the interference of background movements than the OFDM approach.

RF Based Gesture Recognition: Radio Frequency (RF) signals, such as Wi-Fi signals, reflected by human bodies can be used for human gesture and activity recognition [23–28]. However, as the propagation speed of light is almost one million times faster than the speed of sound, it is very difficult to achieve fine-grained distance measurements through RF signals. Therefore, existing Wi-Fi signal based gesture recognition systems cannot perform fine-grained quantification of gesture movement. Instead, they recognize predefined gestures, such as punch, push, or sweep [27,29,30]. When using narrow band RF signals lower than 5 GHz, the state-of-the-art tracking systems have a measurement accuracy of several cm [31, 32]. To the best of our knowledge, the only RF based gesture recognition systems that achieve mm-level tracking accuracy are mTrack [2] and Soli [1], which uses 60 GHz RF signals. The key advantage of our system over mTrack and Soli is that we use speakers and microphones that already exist on most mobile devices to perform device-free tracking of a hand/finger.

Vision Based Gesture Recognition: Vision based gesture recognition systems use cameras or light sensors to capture fine-grained gesture movements [3, 4, 33–35]. For example, Okuli achieves a localization accuracy of 7 mm using LED and light sensors [3]. However, such systems have a limited viewing angle and are susceptible to lighting condition changes [3]. In contrast, LLAP can operate while the device is within the pocket.

3. MEASURE 1-D RELATIVE DISTANCE

In this section, we present our approach to measuring the one-dimensional relative movement distance of a hand/finger, which consists of three steps. First, we use a coherent detector to *down convert* the received sound signal into a complex-valued baseband signal. Second, we measure the path length change based on the phase changes of the baseband signal. Third, we combine the phase changes at different frequencies to mitigate the multipath effect. Before we introduce these three steps, we analyze the limitations of the Doppler shift based approach, which is used by most existing sound-based gesture recognition systems [8, 9, 11–13] and present the advantages of our phase based approach over the Doppler shift based approach.

3.1 Limitations of Doppler Shift Based Distance Measurement

As a moving object changes the frequency of the sound waves reflected by it, by measuring the frequency changes in the received sound signal, which is called Doppler shift, we can calculate the movement speed of the object. The traditional Doppler shift measurement approach, which uses Short-Time Fourier Transform (STFT) to get the Doppler shift, is not suitable for device-free gesture recognition due to its low resolution and highly noisy results.

First, *the resolution of STFT is limited by the fundamental constraints of time-frequency analysis* [36]. The STFT approach first divides the received sound data into data segments, where each segment has equal number (say 2,048) of signal samples, and then performs Fast Fourier Transform (FFT) on each segment to get the spectrum of the given data segment. With a small segment size, the frequency resolution is very low. For example, when the segment size is 2,048 samples and the sampling rate is 48 kHz, the frequency resolution of STFT is 23.4 Hz. This corresponds to a movement speed of 0.2 meters per second (m/s) when the sound wave has a frequency of 20 kHz. In other words, the hand must move at a speed of at least 20 cm per second to be detectable by the STFT approach. Note that improving the frequency resolution is always at the cost of reducing the time resolution [36]. For example, if we use a larger segment size with 48,000 samples to get the frequency resolution of 1 Hz, this will inevitably reduce the time resolution of STFT to one second as it takes one second to collect 48,000 samples when the sampling rate is 48 kHz. Distance measuring schemes with such a low time resolution are unacceptable for interactive inputs because they can only measure the moving distances of a hand/finger at a one-second time interval. Note that the resolution for STFT cannot be improved by padding short data segments with zeros and perform FFT with a larger size, as done in [13], because zero padding is equivalent to convolution with a *sinc* function in the frequency domain. Figure 2 shows the STFT result for a hand that first moves toward and then moves away from the microphone, where each sample segment contains 2,048 samples and is padded with zeros to perform FFT with size of 48,000. Although the frequency resolution seems to be improved to 1 Hz when we perform FFT with a larger size, the high energy band in the frequency domain (red part in the spectrogram) still spans about 80 Hz range, instead of being around 1 Hz. Most of the small frequency variations are buried in this wide band and we can only roughly recognize a positive frequency shift from 4 to 5.2 seconds and a negative frequency shift from 6 to 7.5 seconds.

Second, *Doppler shift measurements are subject to high noises* as shown in Figure 2. In *device-based* tracking systems, such as AAMouse [13], where the sound source or sound receiver is moving, it is possible to use the frequency that has the maximal energy to determine the Doppler shift. In *device-free* tracking systems,

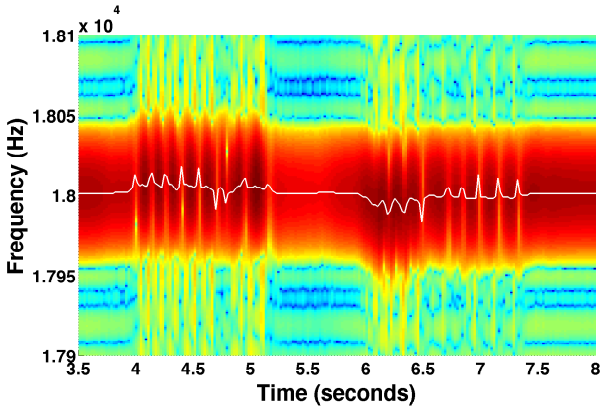


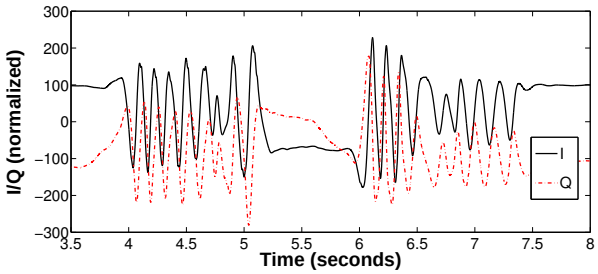
Figure 2: Doppler shift of hand movements

however, the frequency with the highest energy, which is plotted as the white line around 18 kHz in Figure 2, does not closely follow the hand movement because the sound waves reflected by the moving hand are mixed with the sound waves traveling through the Line-Of-Sight (LOS) path as well as those reflected by static objects. Furthermore, there are impluses in the Doppler shift measurements due to *frequency selective fading* caused by the hand movement, *i.e.*, the sound waves traveling from different paths may get cancelled with each other on the target frequency when the hand is at certain positions.

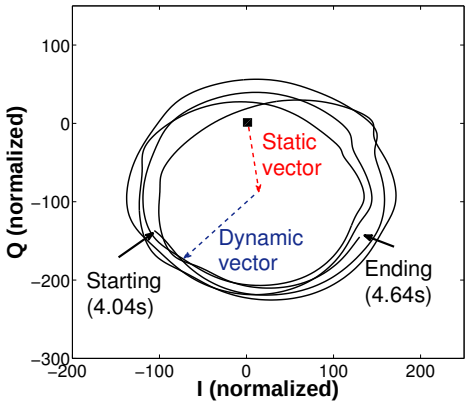
3.2 Phase Based Distance Measurement

Because of the above limitations of Doppler shift based distance measurement, we propose a phase based distance measurement approach for sound signals. As Doppler shift in the reflected signal is caused by the increase/decrease in the phase of the signal when the hand moves close/moves away, the idea is to treat the reflected signal as a *phase modulated* signal whose phase changes with the movement distance. Except for fingerIO that uses OFDM phase [5], no prior work has used phase changes of sound signals to measure movement distance, although the phase of RF baseband signal has been used for measuring the movement distance of objects [2, 23]. Compared to the Doppler shift, the phase change of the baseband signal can be easily measured in the time domain. Figure 3 shows the In-phase (I) and the Quadrature (Q) components of the baseband signal obtained from the same sound record that produces the spectrogram in Figure 2. From Figure 3(a), we observe that the I/Q waveforms remain static when the hand is not moving and vary like sinusoids when the hand moves. Combining the in-phase (as the real part) and quadrature (as the imaginary part) components into a complex signal, we can clearly observe patterns caused by hand movement. Figure 3(b) shows how the complex signal changes during a short time period from 4.04 to 4.64 seconds while the hand moves towards the microphone. We observe that the traces of the complex signal are close to circles on the complex plane.

In essence, the complex signal is a combination of two vectors in the complex plane: we call a *static vector* and a *dynamic vector*. The static vector corresponds to the sound wave traveling through the LOS path or reflected by static objects, such as walls and tables. This vector remains quasi-static during this short time period. The dynamic vector corresponds to the reflection caused by the moving hand. When the hand moves towards the microphone, we observe an increase in the phase of the dynamic vector, which is caused by the decrease in length of the reflected path. As the phase of the signal increases by 2π when the path length decreases by one wavelength of the sound wave, we can calculate the distance that the hand moves via the phase change of the dynamic vector. As-



(a) I/Q waveforms



(b) Complex I/Q traces

Figure 3: Baseband signal of sound waves

suming that the speed of sound is $c = 343$ m/s, the wavelength of sound signals with frequency $f = 18$ kHz is 1.9 cm. We observe that the complex signal moves by about 4.25 circles, which corresponds to an 8.5π increase in phase values in Figure 3(b). Thus, the path length changes by $1.9 \times 4.25 = 8.08$ cm during the 0.6 second shown in Figure 3(b). This is equivalent to hand movement distance of 4.04 cm considering the two-way path length change. Furthermore, we can determine whether the hand is moving toward or moving away from the microphone by the sign of the phase changes. Note that it is important to use both the I and Q components because the movement direction information is lost when we only use a single component or the magnitude [23].

This phase based distance measurement approach has three advantages over the Doppler shift based approach. First, the *accuracy is much higher* because by directly measuring the phase changes, we eliminate the noise-prone steps of first measuring the Doppler shift and then integrating the Doppler shift to get the distance changes. Second, the *latency is much lower* because the phase measurement can be conducted on a short data segment with only hundreds of samples. Third, the speed resolution is much higher because the phase measurement can track small phase changes and slow phase shifts. For example, phase based measurement can easily achieve 2.4 mm distance resolution, which corresponds to a phase change of $\pi/4$ when the wavelength is 1.9 cm. Furthermore, the information is much richer because phase measurements provide more information than what we get from STFT. For example, the phase difference at different frequencies can be used for localizing the hand as discussed in Section 4.

3.3 LLAP Overview

We now give an overview of LLAP when operating on a single sound frequency. Without loss of generality, we assume that the sampling frequency of the device is 48 kHz. We have tested our implementation under other sampling frequencies, *e.g.*, 44.1 kHz,

and obtained similar results as in 48 kHz. LLAP uses Continuous Wave (CW) signal of $A\cos 2\pi ft$, where A is the amplitude and f is the frequency of the sound, which is in the range of $17 \sim 23$ kHz. CW sound signals in this range can be generated by many COTS devices without introducing audible noises [6].

We use the microphones on the same device to record the sound wave using the same sampling rate of 48 kHz. As the received sound waves are transmitted by the same device, there is no Carrier Frequency Offset (CFO) between the sender and receiver. Therefore, we can use the traditional coherent detector structure as shown in Figure 4 to down convert the received sound signal to a baseband signal [37]. The received signal is first split into two identical copies and multiplied with the transmitted signal $\cos 2\pi ft$ and its phase shifted version $-\sin 2\pi ft$. We then use a Cascaded Integrator Comb (CIC) filter to remove high frequency components and decimate the signal to get the corresponding In-phase and Quadrature signals.

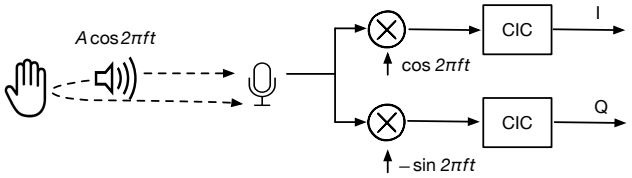


Figure 4: System structure

3.4 Sound Signal Down Conversion

Our CIC filter is a three section filter with the decimate ratio of 16 and differential delay of 17. Figure 5 shows the frequency response of the CIC filter. We select the parameters so that the first and second zeros of the filter appear at 175 Hz and 350 Hz. The pass-band of the CIC filter is $0 \sim 100$ Hz, which corresponds to the movements with a speed lower than 0.95 m/s when the wavelength is 1.9 cm. The second zero of the filter appears at 350 Hz so that the signals at $(f \pm 350)$ Hz will be attenuated by more than 120 dB. Thus, to minimize the interferences from adjacent frequencies, we use a frequency interval of 350 Hz when the speaker transmits multiple frequencies simultaneously. To achieve better computational efficiency, we do not use a frequency compensate FIR filter after the CIC.

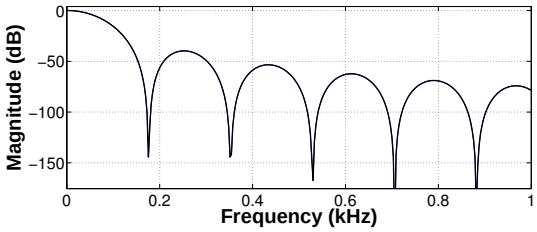


Figure 5: Frequency response of CIC filter

CIC filter incurs low computational overhead as they involve only additions and subtractions. Therefore, we only need two multiplications per sample point for the down conversion, *i.e.*, multiplying the $\cos 2\pi ft$ and $-\sin 2\pi ft$ with each received sample. For 48 kHz sampling rate, this only involves 96,000 multiplications per second and can be easily carried out by mobile devices. After the down conversion, the sampling rate is decreased to 3 kHz to make subsequent signal processing more efficient.

To understand the digital down conversion process, we consider the sound signal that travels through a path p with time-varying path length of $d_p(t)$. This received sound signal from path p can be represented as $R_p(t) = 2A'_p \cos(2\pi ft - 2\pi f d_p(t)/c - \theta_p)$, where $2A'_p$ is the amplitude of the received signal, the term $2\pi f d_p(t)/c$ comes from the phase lag caused by the propagation delay of $\tau_p = d_p(t)/c$ and c is the speed of sound. There is also an initial phase θ_p, which is caused by the hardware delay and phase inversion due to reflection. Based on the system structure shown in Figure 4, when we multiply this received signal with $\cos(2\pi ft)$, we have:

$$\begin{aligned} & 2A'_p \cos(2\pi ft - 2\pi f d_p(t)/c - \theta_p) \times \cos(2\pi ft) \\ = \;& A'_p\big(\cos(-2\pi f d_p(t)/c - \theta_p) + \cos(4\pi ft - 2\pi f d_p(t)/c - \theta_p)\big). \end{aligned}$$

Note that the second term has a high frequency of $2f$ and will be removed by the low-pass CIC filter. Therefore, we have the I-component of the baseband as $I_p(t) = A'_p \cos(-2\pi f d_p(t)/c - \theta_p)$. Similarly, we get the Q-component as $Q_p(t) = A'_p \sin(-2\pi f d_p(t)/c - \theta_p)$. Combining these two components as real and imaginary part of a complex signal, we have the complex baseband as follows, where $j^2 = -1$:

$$B_p(t) = A'_p e^{-j\left(2\pi f d_p(t)/c + \theta_p\right)}. \quad (1)$$

Note that the phase for path p is $\phi_p(t) = -(2\pi f d_p(t)/c + \theta_p)$, which changes by 2π when $d_p(t)$ changes by the amount of sound wavelength $\lambda = c/f$.

3.5 Phase Based Path Length Measurement

As the received signal is a combination of the signals traveling through many paths, we need to first extract the baseband signal component that corresponds to the one reflected by the moving hand so that we can infer the movement distance from the phase change of that component, as we will show next. Thus, we need to decompose the baseband signal into the static and dynamic vector. Recall that the static vector comes from sound waves traveling through the LOS path or the static surrounding objects, which could be much stronger compared to the sound waves reflected by hand. In practice, this static vector may also vary slowly with the movement of the hand. Such changes in the static vector are caused by the blocking of other objects by the moving hand or slow movements of the arm. It is therefore challenging to separate the slowly changing static vector from the dynamic vector caused by a slow hand movement. Existing work in 60 GHz technology uses two methods, Dual-Differential Background Removal (DDBR) and Phase Counting and Reconstruction (PCR), to remove the static vector [2]. However, the DDBR algorithm is susceptible to noises and cannot reliably detect slow movements, while PCR has long latency and requires strong periodicity in the baseband signal. Thus, both of these algorithms are not suitable for our purpose.

We use a heuristic algorithm called *Local Extreme Value Detection* (LEVD) to estimate the static vector. This algorithm operates on the I/Q component separately to estimate the real and imaginary parts of the static vector. The basic idea of LEVD is inspired by the well-known Empirical Mode Decomposition (EMD) algorithm [38]. We first find alternate local maximum and minimum points that are different more than an empirical threshold Thr, which is set as three times of the standard deviation of the baseband signal in a static environment. These large variations in the waveform indicate the movements of surrounding objects. We then use the average of two nearby local maxima and minima as the estimated value of the static vector. Since the dynamic vector has a trace similar to circles, the average of two extremes would be close to the center. Figure 6 shows the LEVD result for a short piece of waveform in Figure 3(a). LEVD pseudocode is in Algorithm 1.

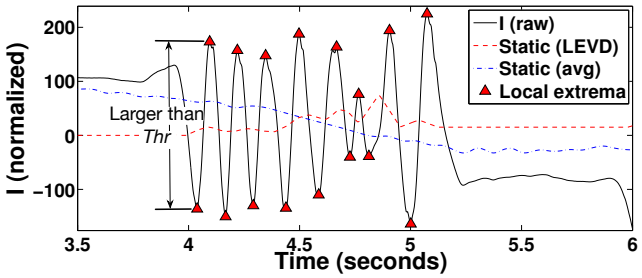


Figure 6: Local extrema based static vector estimation

The advantage of LEVD lies in its robustness to movement speed changes. On one hand, by following the averages of the extreme points, it can quickly trace static vector changes caused by arm movements when the hand moves fast. On the other hand, the estimated vaule of the static vector remains constant when there are no movements or the movements are slow. For example, during the time period of 5.5 to 6 seconds in Figure 6, the normalized value of in-phase component is around -100, which is far away from the actual real part of the static vector. If we use a long term averaging algorithm to estimate the static vector, the estimated real part of the static vector will slowly drift towards -100. In contrast, the static vector estimation of LEVD keeps stable as there are no valid extreme points during this period.

After finding the static vector using LEVD, we subtract it from the baseband signal to get the dynamic vector. We then use the phase $\phi_d(t)$ of the dynamic vector to determine the path length change. We first unwrap the phase $\phi_d(t)$ and the path length change during the time period $0 \sim t$ is given by:

$$d(t) - d(0) = -\frac{\phi_d(t) - \phi_d(0)}{2\pi} \times \lambda \tag{2}$$

where $d(t)$ is the path length from the speaker reflected through the hand to the microphone, and $\lambda = c/f$ is the sound wavelength. When the hand and the microphone/speaker are on the same line, the movement distance of the hand is $(d(t) - d(0))/2$ when it moves towards the speaker, as shown in Figure 4. Note that the distance calculation can be made on a small data segment, *e.g.*, segments with only hundreds of samples. This allows us to respond to hand movements with very low latency, such as 15 ms.

3.6 Multipath Effect Mitigation

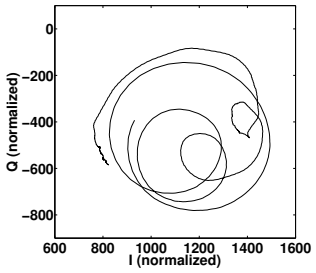
Although LEVD can mitigate the effect of static multipaths by subtracting the static vector, there are dynamic multipaths when the hand moves. A path that the sound wave travels is called static if its length does not change as the hand moves and dynamic if its length changes as the hand moves. An example dynamic path is from the speaker to the hand, and then to a nearby table, and finally to the microphone. Therefore, sometimes there are multiple dynamic vectors and these dynamic vectors may have different phases. This will result in complex signal trajectories, as shown in Figure 7(a). Because of dynamic multipaths, it is difficult to determine the actual phase change from superimposed dynamic vectors.

We use frequency diversity to mitigate the multipath effect. The wavelengths of different sound frequencies are different. Thus, the phases of the same multipath component are different under different frequencies, and the phase changes under different frequencies are also different. The dynamic vectors at different frequencies are combinations of the same set of dynamic paths under different phase offsets. As the multipath components are combined differently in different frequencies, we can combine the measurements

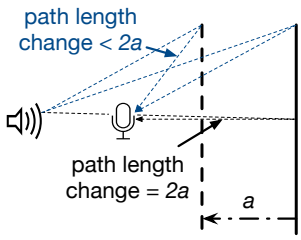
Algorithm 1: Local Extreme Value Detection Algorithm

Input: One baseband signal component $X(t) = I(t)$ or $Q(t)$, $t = 0 \ldots T$
Output: Real or imaginary part of the estimated static vector $S(t)$, $t = 0 \ldots T$

Initialize n: number of extrema, $S(0)$: initial estimation
  $E(n)$: extrema list
for $t = 1$ *to* T **do**
  /*Find extreme points that meet our requirements*/
  if $X(t)$ *is a local maxima or minima* **then**
    Compare $X(t)$ with the last extreme point $E(n)$ in the list;
    if *Both* $X(t)$ *and* $E(n)$ *are local maxima/minima, and the value of* $X(t)$ *is larger/smaller than* $E(n)$ **then**
      $E(n) \leftarrow X(t)$;
    end
    if *One of* $X(t)$ *and* $E(n)$ *is maxima and the other is minima, and* $|X(t) - E(n)| > Thr$ **then**
      $n \leftarrow n + 1$;
      $E(n) \leftarrow X(t)$;
    end
  end
  /*Update the static component estimation using exponential moving average*/
  $S(t) \leftarrow 0.9 \times S(t-1) + 0.1 \times (E(n-1) + E(n))/2$;
end
return $S(t)$



(a) I/Q trace under multipath



(b) Impact of the hand size

Figure 7: Multipath effect

obtained from different frequencies to mitigate the multipath effect. To get the baseband signal at different frequencies, we transmit sounds at multiple frequencies at the same time. The coherent detection structure can be applied on each frequency to obtain one complex baseband signal for each frequency. We remove the interference between adjacent frequencies by carefully selecting the parameters of the CIC filter and the frequency interval. Thus, each frequency can be measured independently. After getting the phase of dynamic vectors at different frequencies, we can obtain the distance change curve over time using the wavelength corresponding to each frequency. We combine the results of different frequencies using linear regression. Our approach is based on two observations. First, the measured distance change should be the same for all frequencies when there is no multipath effect. Second, the distance should change linearly during a short time period, *e.g.*, 10 ms, as the movement speed is almost constant during that short period. Therefore, we use linear regression to find the best line that fits all distance change curves obtained from different frequencies. For those frequencies that have abnormal distance estimation results due to multipath effects, the regression error will be large. We then remove frequencies with large regression errors to achieve a better linear regression result using the rest of frequencies.

3.7 The Impact of Hand Size

The size of the moving object, *i.e.*, the human hand, cannot be ignored when it is close to the speakers and microphones. Human hands have an average length of 15 cm [39]. Thus, different parts of the hand have significant differences in path lengths when we aim at mm-level measurement accuracy. As shown by Figure 7(b), when the hand moves by a distance of a, the path reflected by the center of the hand has path length change of $2a$. However, path reflected by the top of the hand will have smaller path length change, especially when the hand is close to the microphone. As the dynamic vector in the received signal is a mixture of all paths reflected by the hand, the measured path length change will be smaller than the expected value. In our experiments, this type of error increases when the hand is closer to the microphone. As shown by our experiments in Section 6.2, when the hand is 20 cm away from the microphone, the distance measurement error is 3.5 mm; when this distance reduces to 5 cm, the measurement error increases to 6.8 mm. Errors are mostly caused by the impact of the hand size as we consistently underestimates the movement distance. Note that such small error can be compensated by the user when we provide realtime feedbacks to the user. Therefore, we do not use a special algorithm to compensate the underestimation.

4. MEASURE 2-D ABSOLUTE DISTANCE

In this section, we present our 2-D tracking algorithm using sound signals. We first use a delay profile based method to determine the path length so that we can obtain a coarse-grained hand position. We then combine the coarse-grained hand position with the fine-grained path length change to enable 2-D tracking.

4.1 Delay Profile Based Path Measurement

The phase based algorithm in Section 3 only measures the path length change, which is not sufficient for 2-D tracking for two reasons. First, we cannot determine the movement direction only using the path length change due to the lack of the initial position. The path length change is determined by both the movement distance and the movement direction with respect to the speaker and microphone. Movements that are perpendicular to the line connecting the speaker and the microphone incur different changes in path length than movements that are parallel to the line, even if the object moves the same distance. Second, the measurement errors in the path length change accumulate over time. Thus, even if we have the initial hand position, the path length estimation will drift away after tracking for a long time.

In this paper, we propose a delay profile based method to obtain a coarse-grained path length estimation. Our method uses unmodulated CW sound signals to avoid audible noises, such as bursty pulses, introduced by traditional ranging signals. Although the accuracy of the coarse grained measurement is low, which is around 4 cm as shown by our experiments, it serves well for the purpose of providing an initial position, as the realtime tracking is carried out by fine-grained path length change measurements with accuracy at mm-level once the initial position is given.

To measure the path length, we transmit sound signals at N different frequencies $f_k = f_0 + k\Delta f$, $k = 0, \ldots, N-1$, which are separated by a constant frequency interval of Δf. Thus, the baseband signal for any path p at frequency f_k is:

$$B_p(k,t) = A'_{p,k} e^{-j\left(2\pi(f_0+k\Delta f)d_p(t)/c + \theta_{p,k}\right)}. \quad (3)$$

We observe that for a given path length of $d_p(t)$, the phases of the baseband signals at different frequencies decrease as a linear function of Δf, *i.e.*, $-2\pi k\Delta f d_p(t)/c$. Therefore, $B_p(k,t)$ at a given time t will have a constant phase change along the frequency axis, *i.e.*, changing the value of k. If we perform the Inverse Discrete Fourier Transform (IDFT) on $B_p(k,t)$, we have the IDFT result as follows:

$$b_p(n,t) = \frac{1}{N}\sum_{k=0}^{N-1} B_p(k,t) e^{j2\pi kn/N}, n = 0, \ldots, N-1.$$

Suppose we ignore the changes in $A'_{p,k}$ and $\theta_{p,k}$ for this moment, by setting $A'_{p,k} = A'_p$ and $\theta_{p,k} = 0$. In the case that $d_p(t) = \hat{n}c/(N\Delta f)$ for an integer $\hat{n} \in [0, N-1]$, we derive that $b_p(n,t) = A'_p e^{-j2\pi f_0 d_p(t)/c} \times \delta(n-\hat{n},t)$, where $\delta(n,t)$ is the unit impulse function with $\delta(n,t) = 1$, when $n = 0$. For other cases, we have $\delta(n,t) = 0$.

The IDFT of $B_p(k,t)$, denoted as $b_p(n,t)$, is actually a time-delay profile for path p. It has a single peak at time $\hat{n} = Nd_p(t)\Delta f/c$. Therefore, the $\hat{n}$ that maximizes the magnitude of $b_p(n,t)$ indicates the time-delay of path p. Note that both the digital down conversion process and the IDFT operation are linear operations. Therefore, as the received signal is a linear combination of sound waves traveling from different paths, the resulting IDFT is also a linear combination for the delay profile of all paths. As the static vector has been removed by our LEVD algorithm, the IDFT of the dynamic vector contains only the time-delay profile of the moving objects. We identify the peaks in $b_p(n,t)$ and each peak corresponds to one path caused by one moving object. Measuring the delay $\hat{n}$ of the peak gives the path length of the corresponding object. Figure 8 shows the IDFT result $b_p(n,t)$ for a moving hand with $N = 16$ sound frequencies. The "hot" positions indicates the delay profile of high energy sound reflections. There is only one "hot" curve in Figure 8, which corresponds to the dominating reflection path of the hand. We can also measure how the path length changes with time in Figure 8. We observe that the hand starts close to the phone, where the path has a length of about 15 *cm*. As the hand moves away, the corresponding path length increases. We observe that the reflection becomes weak when the hand is about 45 cm away, where the path length increases to 90 cm between 0.7∼1.5 seconds. We also observe that the hand then moves close to the phone twice at 2.9 and 6 seconds.

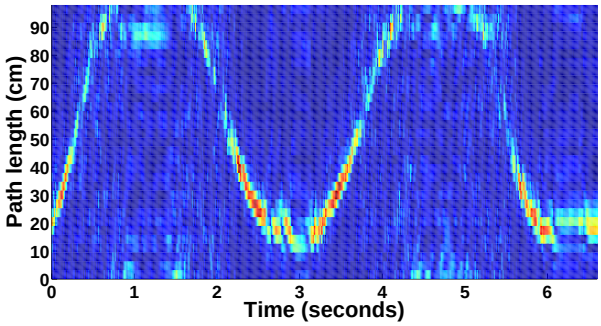


Figure 8: Delay profile $b_p(n,t)$ for a moving hand

4.2 Parameter Setting

The time-delay profile measurement has two parameters that need to be carefully chosen: the frequency interval Δf and the number of frequencies N. For Δf, on one hand, Δf should be large enough so that we can separate high speed movements at adjacent carrier frequencies. For example, a movement with a speed of 1 m/s leads to frequency components around 100 Hz in the baseband signal. Thus, adjacent frequencies should be separated by at least 200 Hz. On the other hand, Δf should be small enough so that we can avoid time-domain aliasing. Note that Δf determines the

time domain aliasing range. The estimated peak position $\hat{n}$ is given as an integer value modulo N, which is in the range of $0 \sim N-1$. Therefore, a reflector with path length of d will have the same time-delay profile as those with path length of $d + m\,c/\Delta f$, where m is an integer. For example, when Δf is 350 Hz, paths with length of 0 cm will have the same delay profile as paths with length of $c/\Delta f = 98$ cm. Such time domain aliasing can be observed in Figure 8, where the high energy curve wraps back to around 0 cm when the path length is larger than 98 cm between $4.5 \sim 5.1$ seconds. As we aim at an operational range of less than 50 cm, we let Δf to be 350 Hz. For the number of frequencies N, on one hand, a larger N gives us a better distance resolution because a larger N leads to a smaller path length difference $c/(N\Delta f)$ between two adjacent points $\hat{n}$ and $\hat{n}+1$. On the other hand, a larger N requires higher bandwidth and reduces the energy that we can transmit in a single frequency. As the total bandwidth for N frequencies is $(N-1)\Delta f$, we can only fit a limited number of frequencies into the available frequency range, *e.g.*, $17 \sim 23$ kHz. Furthermore, the more frequencies we use, the less energy we can transmit in each frequency because the total energy that can be transmitted by the speaker is limited for mobile devices. When the transmission energy in each frequency is reduced, the Signal-to-Noise Ratio (SNR) is also reduced and the phase measurement becomes less reliable. In this paper, we let $N = 16$, which implies that the bandwidth is 5.25 kHz and the path length resolution is 6.16 cm. The actual path length measurement error is smaller than 4 cm when the target is within 30 cm to the phone, as in our experimental results on Section 6.2.

4.3 System Calibration

The initial phase offset $\theta_{p,k}$ comes from two sources: one is the phase inversion caused by reflection, which is the same for all frequencies, and the other is the delay in audio playing and recording process caused by the hardware limitation of the mobile device, which is different for different device models. Because of the delay, the time that we transmit the CW to the speaker is misaligned with the reference $\cos(2\pi ft)$ signal that is used for multiplication in the coherent detector. Thus, there is a random offset of Δt between the emitted and received signal. Consequently, there will be a time offset of Δt in $b_p(n,t)$ after the IDFT. This time offset, whose value depends on the audio initialization process, will remain constant after the system starts emitting and receiving continuous signals.

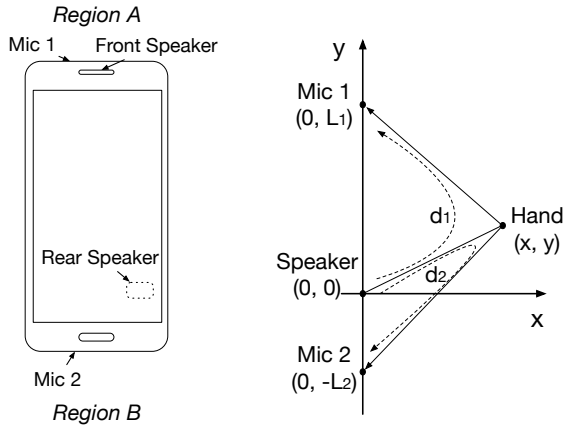
We perform the time offset calibration after the system starts emitting sound signals. As the hardware/operating system introduced time offset Δt is the same for all paths, we use the LOS path as the reference path in our calibration process. As we know the exact distance between the speaker and microphone for a given mobile device model, we can calculate the expected $\hat{n}_{LOS}$ for the LOS path. As the static vector is dominated by the LOS path when there are no large reflectors around, if we perform IDFT on the *static* vector of different frequencies, we expect the highest peak will appear at $\hat{n}_{LOS}$ if $\Delta t = 0$. If we observe that the peak is not at $\hat{n}_{LOS}$, we apply a delay $\Delta t'$ on the reference $\cos(2\pi ft)$ signal and iteratively adjust the value of $\Delta t'$ until the peak appears at the expected position. In our implementation, the average time used for the calibration process is 0.82 seconds with a standard deviation of 0.16 seconds.

4.4 Combining Fine-grained Phase and Coarse-grained Delay Measurements

Our 2-D tracking requires both the fine-grained phase measurement and the coarse-grained delay profile measurement. The phase measurement provides accurate and realtime distance changes so that the system can respond to user actions with high accuracy and low latency. The delay profile measurement gives the estimation of path length so that the error in phase measurements would not accumulate over time. We combine the fine-grained and coarse-grained measurements to achieve both low latency and stableness in measurements. From Figure 8, we observe that the delay profile gives consistent estimations when the energy of the reflected sound is high, *e.g.*, between 2.1∼2.5 seconds. Therefore, we use the delay profile based path length estimation only when there is a dominating peak in $b_p(n,t)$ that has normalized energy higher than a given threshold. In such cases, we augment the path length estimation obtained through the delay profile with the path length traced through the phase measurements using an Exponential Moving Average (EMA) algorithm. If the hand reflection is weak and there is no dominating peak in $b_p(n,t)$, we only use the phase change to update the path length as the delay profile is unreliable.

4.5 2-D Gesture Tracking

The position of the hand is determined through multiple path length measurements obtained from different speaker/microphone pairs on the mobile device. Figure 9(a) shows the positions of the speakers and microphones on a typical mobile phone, Samsung Galaxy S5. To measure the path length for multiple speakers/microphones, we use stereo playback and recording capability that is available on many mobile devices. For example, we can record the sound at two microphones that are located at different positions to get two path measurements at the same time. When there are multiple speakers, we can separate the signal from different speakers by assigning different frequencies to each speaker.



(a) Layout of Samsung S5 (b) Geometric abstraction

Figure 9: Two dimensional tracking

To simplify our discussion, let us consider a mobile phone with one speaker and two microphones, as shown in Figure 9(b). Consider the case where the speaker is placed at the origin, while the two microphones have coordinates of $(0, L_1)$ and $(0, -L_2)$, respectively. Suppose that the path length from the speaker through the hand to two microphones are d_1 and d_2, respectively. The coordinates (x, y) of the hand should be on the ellipses defined by:

$$\frac{4x^2}{d_1^2 - L_1^2} + \frac{4(y - L_1/2)^2}{d_1^2} = 1 \tag{4}$$

$$\frac{4x^2}{d_2^2 - L_2^2} + \frac{4(y + L_2/2)^2}{d_2^2} = 1 \tag{5}$$

Solving this we have:

$$x = \frac{\sqrt{(d_1^2 - L_1^2)(d_2^2 - L_2^2)\left((L_1 + L_2)^2 - (d_1 - d_2)^2\right)}}{2(d_1 L_2 + d_2 L_1)}$$
$$y = \frac{d_2 L_1^2 - d_1 L_2^2 - d_1^2 d_2 + d_2^2 d_1}{2(d_1 L_2 + d_2 L_1)} \quad (6)$$

As the distance L_1 and L_2 between the speaker to the microphones are fixed for a given device, we can directly calculate the position of the hand using the path length d_1 and d_2.

The pseudocode of our 2-D tracking algorithm is in Algorithm 2. This algorithm uses the path length estimation on two microphones to track the hand. Note that it is possible to use sophisticated tracking algorithms, such as Kalman filters, to further improve the tracking performance. We choose not to use them in our implementation because they incur high computational cost. However, for mobile devices with enough computational power, we recommend using them.

Algorithm 2: Two Dimensional Tracking Algorithm

```
Input: Data segment of baseband signal for two microphones on N
       frequencies
Output: Updated hand position
foreach microphone do
    foreach frequency do
        Estimate the static vector using LEVD;
        Obtain the dynamic vector by subtracting the static vector
        from the baseband signal;
        Calculate the path length change based on the phase change
        of the dynamic vector;
    end
    Use linear regression to combine the path length change
    estimation in different frequencies;
    Update the path length using the path length change estimation;
    Take IDFT of the dynamic vector of different frequencies to get
    b_p(n, t);
    if Peak value in b_p(n, t) is larger than threshold then
        Estimate the coarse-grained path length using n̂;
        Use EMA to augment the coarse-grained estimation;
    end
end
Use the path length of two microphones to update the hand position;
```

5. IMPLEMENTATION

We implemented LLAP on both the Android and iOS platforms. On the Android platform, we implement most signal processing algorithms as C functions using Android NDK to achieve better efficiency. Our implementation works as an APP that can draw the 2D hand traces in realtime on recent Android phones, *e.g.*, Samsung Galaxy S5 with Android 5.0 OS. On the iOS platform, we use the vDSP accelerate framework which achieves much better computational efficiency than the Android platform. However, the iOS platform only supports single channel recording. So, we only implement 1-D hand tracking on the iOS system. Note that we need to reconfigure the system for certain mobile phones, so that the hardware echo cancellation can be bypassed.

There are some limitations in the hardware and operating system of existing mobile phones. First, the placement of the microphones and speakers are not optimized for gesture tracking. For example, the microphones for Samsung S5 are pointing towards opposite directions as shown in Figure 9(a). When the hand is in *Region A* shown in Figure 9(a), the reflected signal obtained by microphone 1 is very good while microphone 2 only gets weak signals. Therefore, to achieve strong signals for both microphones, our 2-D tracking experiments are performed in front of or behind the phone when using the front or rear speaker, rather than in region A or B. Second, the latency of our system is constrained by the operating system. Although LLAP can operate on short data segments, the Android system only returns sound data in 10∼20 ms intervals, depending on the phone models. Therefore, we choose data segment size of 512 samples in our implementation, which has time duration of 10.7 ms when the sampling rate is 48 kHz. The iOS system provides better sound APIs which can operate at data segment sizes as small as 32 samples. However, the iOS system only supports recording from a single microphone so that we did not implement 2-D tracking on the iOS platform. Even with these hardware and software limitations, LLAP achieves good accuracy and latency on existing mobile phones. We believe that if the mobile phones were designed with hardware/software optimizations for sound based gesture tracking, such as placing the speaker and microphones on one side of the phone, the performance of LLAP could be even better.

6. EVALUATION

6.1 Evaluation Setup

We conducted experiments on Samsung Galaxy S5 using its rear speaker, top microphone, and bottom microphones in normal office and home environments with the phone on a table as shown in Figure 10. Experiments were conducted with five human users. The users interacted with the phone using their bare hands without wearing any accessory.

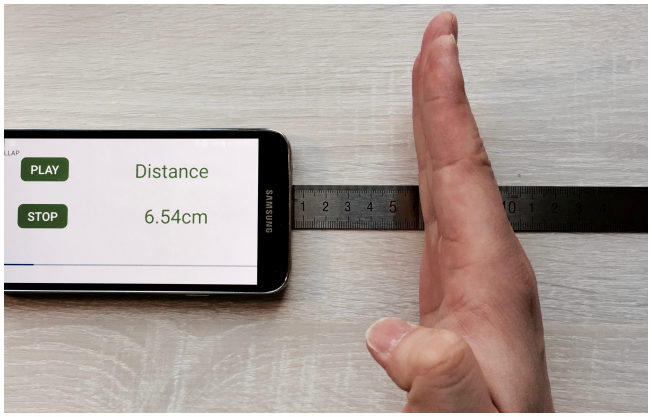
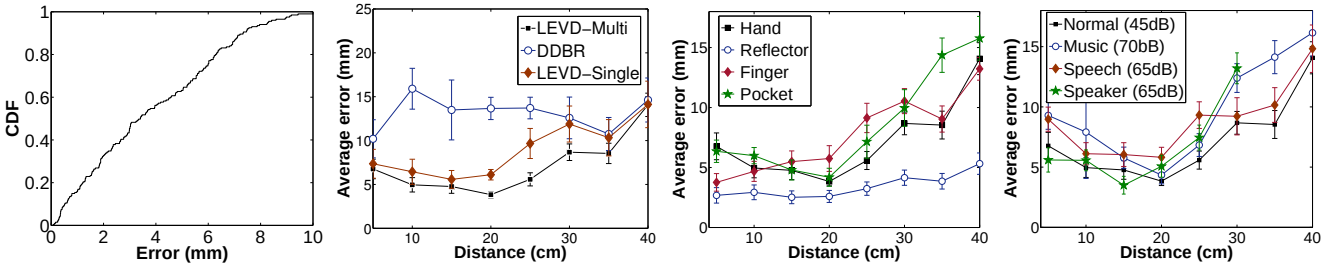


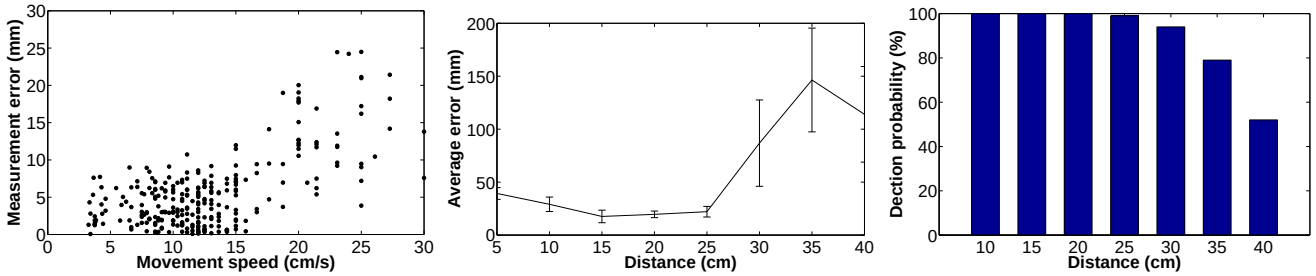
Figure 10: Experimental setup

For *1-D tracking*, we evaluated LLAP using three metrics: (1) Movement distance error: the difference between the LLAP reported movement distance and the ground truth movement distance measured by a ruler placed along the movement path. (2) Absolute path length error: the difference between the LLAP reported path length and the ground truth measured by a ruler. (3) Micro movement detection accuracy: the probability that LLAP correctly detects a small single-finger movement and reports the correct movement direction of either moving towards or away from the phone. For *2-D tracking*, we evaluated LLAP using two metrics: (4) Tracking error: the distance between the LLAP reported trace and the standard drawing template. Because the 2-D tracking error is defined in a different way to 1-D tracking, the results for these two metrics are not directly comparable. (5) Character recognition ac-



(a) CDF for measurement error (b) Different algorithms (c) Different objects (d) Different environments

Figure 11: 1-D Movement distance errors. (Confidence intervals for (b), (c), and (d) are 95%.)



(a) Movement error for different speeds (b) Absolute path length error (c) Micro movement detection accuracy

Figure 12: Micro benchmarks (Confidence interval for (b) is 95%.)

curacy: the probability that the tracking trace reported by LLAP, based on the character drawn by a user, can be correctly recognized by MyScript, a handwriting recognition tool [40]. For *efficiency*, we evaluated LLAP using two metrics: (6) Response latency: the time used by LLAP to accumulate and process the sound data before it responses to the hand movement. (7) Power consumption: the energy consumption of LLAP on mobile phones.

6.2 Experimental Results

LLAP achieves an average movement distance error of 3.5 mm when the hand moves for 10 cm at a distance of 20 cm. We moved the hand in "Region A" in Figure 9(a) and measured the movement distance using the top microphone and the rear speaker. The initial hand position was 20 cm away from the microphone and the hand moved away from the microphone for a distance of 10 cm. Figure 11(a) show the Cumulative Distribution Function (CDF) of the distance measurement error for 200 movements. The 90th percentile measurement error is 7.3 mm and the average error is 3.5 mm as shown in Figure 11(a).

LLAP achieves an average movement distance error of less than 8.7 mm when the hand moves for 10 cm at a distance of less than 35 cm. Figure 11(b) shows the average movement distance error when the hand is at different distances from the microphone in side-by-side comparison with DDBR, the movement distance measurement algorithm proposed in [2]. Results show that our LEVD algorithm outperforms the DDBR algorithm as DDBR is susceptible to noises. Results also show that LEVD with signals of multiple frequencies outperforms LEVD with signals of a single frequency in terms of distance measurement accuracy by 21% on average. We observe that for LEVD, the movement distance error increases when the hand is too close or too far from the microphone. When the hand is too close to the microphone, the impact of the hand size increases, which leads to larger movement distance errors. To verify the impact of hand sizes, we conducted the same set of experiments with different types of moving objects, including a hand, two fingers, and a plastic flat reflector with an area of 12×4 cm. As shown in Figure 11(c), smaller objects, such as two-fingers and the small reflector, result in a better accuracy of 3.76 mm and 2.68 mm, respectively, when the object is very close to the microphone (within a distance of 5 cm). Due to the better reflection ability of the reflector, the measurement accuracy for the reflector at a distance of 40 cm is 5.32 mm, which is much smaller than that of the hand and two-fingers. This is because when the hand is too far from the microphone, the sound signal reflected from the hand is too weak and the SNR is too low, which leads to larger movement distance errors. When the hand is more than 40 cm way from the microphone, the error increases to more than 14 mm. Other small variations in accuracy in Figure 11(c) are mostly caused by the different multi-path conditions at different distances. LLAP can also operate while the device is inside the pocket. Figure 11(c) shows that the measurement error of LLAP only slightly increases by 1.4 mm on average when the device is inside a bag made of cloth.

LLAP is robust to background noises and achieves an average movement distance error of 5.81 mm under noise interferences. Figure 11(d) shows the measurement error under four different environments: the "normal" environment is a typical silent indoor one, the "music" environment is an indoor environment with pop music being played with normal volume, the "speech" environment is a room with people talking at the same time, and the "speaker" environment is playing music from the speaker on the same device. The sound pressure levels measured at these four environments are 45 dB, 70 dB, 65 dB, and 65 dB, respectively. We observe that LLAP has slightly larger movement distance errors under noise interferences. Compared to the "normal" environment, the movement distance errors are increased by 2.45 mm and 1.66 mm (averaged over different distances) for the "music" and "speech" environments, respectively. Because LLAP only uses the narrow baseband signal around each transmitted frequency, the robustness of LLAP under audible sound noises is sufficient for practical usage. For the challenging scenario where the smart phone plays music from the *same* speaker that is used for sending the CW signal, LLAP still achieves distance accuracy of 7.5 mm when the hand is within 25

cm to the speaker. Due to the strong self-interference in this scenario, the measurement error at a distance larger than 30 cm is more than 20 mm. Note that we can still use the microphone for normal recording when LLAP is running. Thus, LLAP do not block the normal operation of the speakers and microphones on the device.

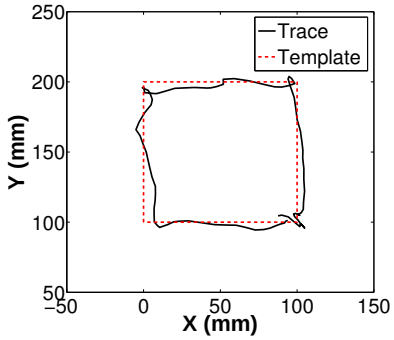
LLAP can reliably measure the movement distance with speeds from 4 cm/s to 25 cm/s. In our experiments, a user moves his hand at different speeds for a distance of 10 cm. Figure 12(a) shows the distribution of the movement distance errors with respect to the movement speed. We observe that for slow movement speeds from 4 cm/s to 15 cm/s, the error distribution is consistent with an average error of 3.64 mm. The error increases when the movement speed is higher than 15 cm/s. The movement distance error of faster movements are higher because the changes in static vector introduced by the arm when the hand moves faster are larger. However, the maximum error is still less than 25 mm. Thus, LLAP can handle both slow and fast hand movements.

LLAP achieves an average absolute path length error of less than 40 mm when the hand is within 25 cm from the phone as shown in Figure 12(b). We placed the hand at different distances to the phone and measure the absolute length of the path reflected by the hand. Within 25 cm to the microphone, the average absolute path length error is 3.57 cm. Note that the absolute path length is the length that the sound signals travels, which is twice of the distance between the phone and the hand.

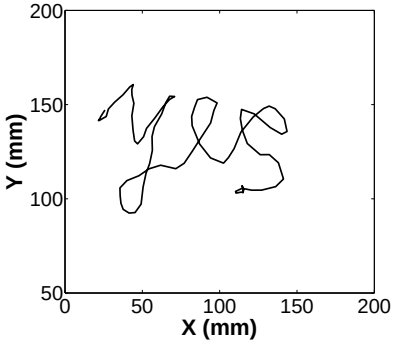
LLAP achieves a micro movement detection accuracy of higher than 94% within a distance of 30 cm. In our experiments, a user moves only the index finger for a distance of 5 mm at different distances from the microphone. We consider the detection to be successful only when LLAP correctly detects the movement *and* gives the correct movement direction for the micro movement. Figure 12(c) shows the micro movement detection accuracy of LLAP. We observe that the detection accuracy is above 94% when the finger is within 30 cm and quickly reduces when the distance is larger than 35 cm due to the weaker signals reflection of the finger and the resulting lower SNR. Results also show that LLAP has low false positive ratios. When placed in a silent environment, LLAP makes only one false detection of movement (with a distance larger than 5 mm) among 35,015 detection decisions. This gives a false positive rate of only 0.003%.

For 2-D tracking, LLAP achieves a tracking error of 4.57 mm. Figure 13 shows samples of square and word drawn by LLAP. These drawings have dimensions around 10×10 cm, which is in the comfortable range for gesture inputs. To evaluate the performance of tracking errors, we request 5 users to draw according to a square template of 10×10 cm using LLAP. The average time for users to finish one drawing is 5.4 seconds. Figure 14(a) shows the average error of the estimated trace to the template, which is defined as the distance of points on the trace to the nearest point on the template. The error is the average result of 50 drawings for each user. The average error of the drawing is $3.34 \sim 5.54$ mm (with a mean of 4.57 mm). The maximum deviation from the template is $13.1 \sim 20.7$ mm (with mean of 16.41 mm) for different users. Note that we adopt a relative distance error measurement and the actual trace of the hand may have an offset to the estimated trace. Since we provide realtime feedback to the user, users can control the drawing trace to follow the template and compensate for small offsets. The tracking performance of LLAP is robust to noises. In the "music" and "speech" environments, the average tracking error is 4.89 and 4.81 mm, respectively.

For 2-D tracking, the characters and words drawn by LLAP can be recognized by MyScript with accuracies of 92.3% and 91.2%, respectively. Figure 14(b) shows the average recognition accuracies for each user. For characters, each user drew at least 5 times for each of the 26 Latin alphabets. For words, each user drew at least 5 times for each word in a list of 11 words, such as "yes", "can" or "bye". For the lower case letters that cannot be drawn with a single stroke, such as "i", we used the upper case letter for these characters. The average character recognition accuracies for different users are in the range of $87.6\% \sim 95.3\%$, with an average accuracy of 92.3% over all users. The average word recognition accuracies for different users are in the range of $88.4\% \sim 94.5\%$, with an average accuracy of 91.2%.



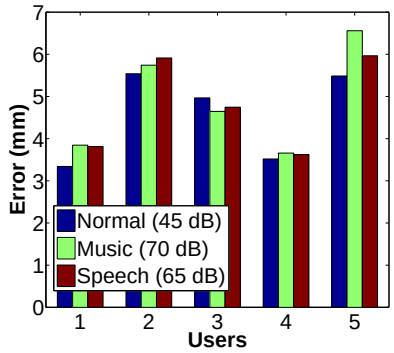
(a) Drawing square



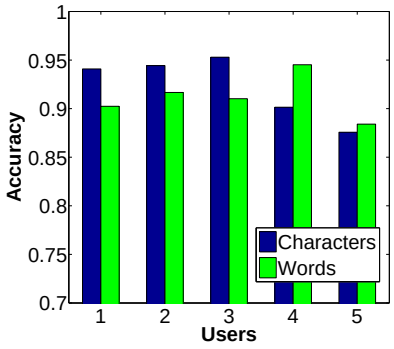
(b) Drawing word

Figure 13: Sample results of drawing in the air

For responsiveness, LLAP achieves a latency of less than 15 ms on mobile phones. We measured the running time for a Samsung S5 with Qualcomm Snapdragon 2.5GHz quad-core CPU to process data segments of 512 samples (time duration of 10.7 ms at 48 kHz sampling rate). The average time used for our algorithms of baseband down conversion, LEVD, phase based path length change measurement, and delay profile based absolute path length measurement are 3.55, 0.17, 0.36, and 0.08 ms, respectively. With processing time for other operations, the total running time for LLAP to process 10.7 ms of data is 4.32 ms, which meets the real time processing requirement. Although we have simplified the down conversion process using CIC filters, most of the processing time (about 82%) is still used by baseband down conversion because we need to process the 48 kHz samples at 16 frequencies for each of the two microphones, which incurs considerable operations. After baseband down conversion, the sampling rate is reduced by 16 times. Therefore, the processing time for rest of the operations becomes significantly smaller. Our implementation on iOS platform has better performance as we use the vDSP accelerate framework for audio processing. The average time for a iPhone 6s with A9 processor to process 512 samples is only 0.30 ms, as shown in Table 1. Although our implementation on iOS is simpler, *e.g.*, only pro-



(a) Tracking error



(b) Recognition accuracy

Figure 14: Performance of 2-D tracking

cessing 16 frequencies on a single microphone with 1-D tracking, the processing speed for down conversion is still much faster than using Android NDK. This shows that the DSP acceleration framework on mobile devices can efficiently fulfill the computational requirements of LLAP. The overall latency of our system is smaller than 15 ms. Therefore, there is no human perceivable delay in the response during our experiments.

Table 1: Time to process audio segment with 10.7 ms duration

Phone	Down Conversion	LEVD	Phase measure	Delay profile	Total
Samsung S5	3.55ms	0.17ms	0.36ms	0.08ms	4.32ms
iPhone 6s	0.19ms	0.03ms	0.08ms	–	0.30ms

LLAP runs for more than 10.5 hours on COTS mobile devices. To measure the energy consumption of LLAP, we run the LLAP application with maximum audio volume and realtime 1-D tracking on an iPhone 6s. A fully charged iPhone 6s can continuously run LLAP for 10.57 hours. The instrument tools provided by Xcode rate the energy usage level of LLAP as 0 (lowest) in the scale of $0 \sim 20$. The reason of good power efficiency is that playing/recording the sound incurs low energy cost and our implementation only consumes less than 3% CPU time on the iOS platform.

7. LIMITATIONS

LLAP demonstrates that commercial mobile devices can use acoustic phase information to track hands/fingers with millimeter-level accuracy. However, our current implementation of LLAP has the following three limitations. First, LLAP can only track a single moving object. Therefore, it treats the finger and the hand as one integrated object. LLAP cannot recognize complex gestures that involves multiple moving fingers, such as "pinch". Furthermore, LLAP cannot detect events such as "touch" as in mTrack [2]. Therefore, the users must finish the drawing in a single stroke. An interesting future research topic would be separating multiple fingers using sound signals recorded by multiple microphones at multiple frequencies. Second, LLAP can be interfered by nearby moving objects, *e.g.*, a person walking within 2 meters or the moving body of the user himself. Therefore, our current implementation works only when the surrounding is relatively static. We consider to remove the background movements using the fact that these movements occur at a longer distance than the gesture so that they appear at a different location in the delay profile. Third, LLAP can be interfered by high frequency noises, especially sounds emitted by other LLAP devices. Such interference can be mitigated by using better speakers and microphones which supports up to 80 kHz frequency [41]. This is because there are less interferences in higher frequencies (*e.g.*, higher than 40 kHz) and different LLAP devices can use different frequency bands as there are more spectrum resources in higher frequency bands.

8. CONCLUSION

We make following key contributions. First, we propose an acoustic phase based gesture tracking algorithm, which achieves millimeter-level accuracy, less than 15 ms latency, and lower than 3% CPU usage on commercial mobile phones. Second, we propose a suite of novel signal processing algorithms, such as the LEVD, the phase based path length change measurement, and the delay profile based path length change measurement algorithms, to enable our device-free approach to gesture tracking using acoustic signals. We implemented our prototype system LLAP on commercial mobile phones and evaluated its performance in various settings for seven metrics. We envision that LLAP will enable a plethora of novel device-free gesture based mobile applications.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and the students/volunteers in our lab who helped in collecting our dataset. This work is partially supported by the National Natural Science Foundation of China under Grant Numbers 61373129, 61472184, 61321491, and 61472185, the National Science Foundation under Grant Numbers CNS-1421407, Collaborative Innovation Center of Novel Software Technology and Industrialization, and the Jiangsu High-level Innovation and Entrepreneurship Program.

9. REFERENCES

[1] Google project soli. https://www.google.com/atap/project-soli/.

[2] Teng Wei and Xinyu Zhang. mTrack: High-precision passive tracking using millimeter wave radios. In *Proc. ACM MobiCom*, 2015.

[3] Chi Zhang, Josh Tabor, Jialiang Zhang, and Xinyu Zhang. Extending mobile interaction through near-field visible light sensing. In *Proc. ACM MobiCom*, 2015.

[4] Leap Motion. https://www.leapmotion.com/.

[5] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. FingerIO: Using active sonar for fine-grained finger tracking. In *Proc. ACM CHI*, 2016.

[6] A Rodrìguez Valiente, A Trinidad, JR García Berrocal, C Górriz, and R Ramírez Camacho. Extended

high-frequency (9–20 kHz) audiometry reference thresholds in 645 healthy subjects. *International journal of audiology*, 53(8):531–545, 2014.

[7] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. Beepbeep: a high accuracy acoustic ranging system using COTS mobile devices. In *Proc. ACM SenSys*, 2007.

[8] Zengbin Zhang, David Chu, Xiaomeng Chen, and Thomas Moscibroda. Swordfight: Enabling a new class of phone-to-phone action games on commodity phones. In *Proc. ACM MobiSys*, 2012.

[9] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. Soundwave: using the doppler effect to sense gestures. In *Proc. ACM CHI*, 2012.

[10] Zheng Sun, Aveek Purohit, Raja Bose, and Pei Zhang. Spartacus: spatially-aware interaction for mobile devices through energy-efficient audio sensing. In *Proc. ACM MobiSys*, 2013.

[11] Md Tanvir Islam Aumi, Sidhant Gupta, Mayank Goel, Eric Larson, and Shwetak Patel. Doplink: Using the doppler effect for multi-device interaction. In *Proc. ACM UbiComp*, 2013.

[12] Ke-Yu Chen, Daniel Ashbrook, Mayank Goel, Sung-Hyuck Lee, and Shwetak Patel. Airlink: sharing files between multiple devices using in-air gestures. In *Proc. ACM UbiComp*, 2014.

[13] Sangki Yun, Yi-Chao Chen, and Lili Qiu. Turning a mobile device into a mouse in the air. In *Proc. ACM MobiSys*, 2015.

[14] Nissanka B Priyantha, Anit Chakraborty, and Hari Balakrishnan. The cricket location-support system. In *Proc. ACM MobiCom*, 2000.

[15] Jie Yang, Simon Sidhom, Gayathri Chandrasekaran, Tam Vu, Hongbo Liu, Nicolae Cecan, Yingying Chen, Marco Gruteser, and Richard P. Martin. Detecting driver phone use leveraging car speakers. In *Proc. ACM MobiCom*, 2011.

[16] Yu-Chih Tung and Kang G Shin. Echotag: Accurate infrastructure-free indoor location tagging with smartphones. In *Proc. ACM MobiCom*, 2015.

[17] Wenchao Huang, Yan Xiong, Xiang-Yang Li, Hao Lin, Xufei Mao, Panlong Yang, and Yunhao Liu. Shake and walk: Acoustic direction finding and fine-grained indoor localization using smartphones. In *Proc. IEEE INFOCOM*, 2014.

[18] Rajalakshmi Nandakumar, Shyamnath Gollakota, and Nathaniel Watson. Contactless sleep apnea detection on smartphones. In *Proc. ACM MobiSys*, 2015.

[19] Junjue Wang, Kaichen Zhao, Xinyu Zhang, and Chunyi Peng. Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization. In *Proc. ACM MobiSys*, 2014.

[20] Tong Zhu, Qiang Ma, Shanfeng Zhang, and Yunhao Liu. Context-free attacks using keyboard acoustic emanations. In *Proc. ACM CCS*, 2014.

[21] Jian Liu, Yan Wang, Gorkem Kar, Yingying Chen, Jie Yang, and Marco Gruteser. Snooping keystrokes with mm-level audio ranging on a single phone. In *Proc. ACM MobiCom*, 2015.

[22] Maotian Zhang, Panlong Yang, Chang Tian, Lei Shi, Shaojie Tang, and Fu Xiao. Soundwrite: Text input on surfaces through mobile acoustic sensing. In *Proc. ACM SmartObjects*, 2015.

[23] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Understanding and modeling of WiFi signal based human activity recognition. In *Proc. ACM MobiCom*, 2015.

[24] Kamran Ali, Alex X. Liu, Wei Wang, and Muhammad Shahzad. Keystroke recognition using WiFi signals. In *Proc. ACM MobiCom*, 2015.

[25] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. Whole-home gesture recognition using wireless signals. In *Proc. ACM MobiCom*, 2013.

[26] Fadel Adib, Zachary Kabelac, and Dina Katabi. Multi-person motion tracking via RF body reflections. In *Proc. Usenix NSDI*, 2015.

[27] Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota. Bringing gesture recognition to all devices. In *Proc. Usenix NSDI*, 2014.

[28] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. E-eyes: In-home device-free activity identification using fine-grained WiFi signatures. In *Proc. ACM MobiCom*, 2014.

[29] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. WiGest: A ubiquitous WiFi-based gesture recognition system. In *Proc. IEEE INFOCOM*, 2015.

[30] Pedro Melgarejo, Xinyu Zhang, Parameswaran Ramanathan, and David Chu. Leveraging directional antenna capabilities for fine-grained gesture recognition. In *Proc. ACM UbiComp*, 2014.

[31] Li Sun, Souvik Sen, Dimitrios Koutsonikolas, and Kyu-Han Kim. WiDraw: Enabling hands-free drawing in the air on commodity wifi devices. In *Proc. ACM MobiCom*, 2015.

[32] Jue Wang, Deepak Vasisht, and Dina Katabi. RF-IDraw: virtual touch screen in the air using RF signals. In *Proc. ACM SIGCOMM*, 2014.

[33] Microsoft Kinect. http://www.microsoft.com/en-us/kinectforwindows/.

[34] Robert Xiao, Chris Harrison, Karl DD Willis, Ivan Poupyrev, and Scott E Hudson. Lumitrack: low cost, high precision, high speed tracking with projected m-sequences. In *Proc. ACM UIST*, 2013.

[35] Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. In-air gestures around unmodified mobile devices. In *Proc. ACM UIST*, 2014.

[36] Leon Cohen. *Time-frequency analysis*. Prentice hall, 1995.

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

[38] Norden E Huang, Zheng Shen, Steven R Long, Manli C Wu, Hsing H Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry H Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 454, pages 903–995. The Royal Society, 1998.

[39] AK Agnihotri, B Purwar, N Jeebun, and S Agnihotri. Determination of sex by hand dimensions. *The Internet Journal of Forensic Science*, 1(2):12–24, 2006.

[40] MyScript. http://myscript.com/.

[41] Knowles Electronics. SPH0641LU4H-1: Digital zero-height SiSonic™micophone with multi-mode and ultrasonic support, 2014.