Watching Your Phone's Back: Gesture Recognition by Sensing Acoustical Structure-borne Propagation

LEI WANG, Peking University, China
XIANG ZHANG, University of New South Wales, Australia and Harvard University, United States
YUANSHUANG JIANG and YONG ZHANG, Shenzhen Institute of Advanced Technology, CAS, China
CHENREN XU, RUIYANG GAO, and DAQING ZHANG, Peking University, China

Gesture recognition on the back surface of mobile phone, not limited to the touch screen, is an enabling Human-Computer Interaction (HCI) mechanism which enriches the user interaction experiences. However, there are two main limitations in the existing Back-of-Device (BoD) gesture recognition systems. They can only handle coarse-grained gesture recognition such as tap detection and cannot avoid the air-borne propagation suffering from the interference in the air. In this paper, we propose StruGesture, a fine-grained gesture recognition system using the back of mobile phones with ultrasonic signals. The key technique is to use the structure-borne sounds (*i.e.*, sound propagation via structure of the device) to recognize sliding gestures on the back of mobile phones. StruGesture can fully extract the structure-borne component from the hybrid Channel Impulse Response (CIR) based on Peak Selection Algorithm. We develop a deep adversarial learning architecture to learn the gesture-specific representation for robust and effective recognition. Extensive experiments are designed to evaluate the robustness over nine deployment scenarios. The results show that StruGesture outperforms the competitive state-of-the-art classifiers by achieving an average recognition accuracy of 99.5% over 10 gestures.

CCS Concepts: • **Human-centered computing** → **Human computer interaction (HCI)**;

Additional Key Words and Phrases: Structure-borne Recognition, Mobile System, Acoustic Sensing

ACM Reference Format:
Lei Wang, Xiang Zhang, Yuanshuang Jiang, Yong Zhang, Chenren Xu, Ruiyang Gao, and Daqing Zhang. 2021. Watching Your Phone's Back: Gesture Recognition by Sensing Acoustical Structure-borne Propagation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 2, Article 82 (June 2021), 26 pages. https://doi.org/10.1145/3463522

1 INTRODUCTION

Gesture recognition is a basic HCI mechanism for electronic devices in Internet of Things (IoT), especially for smartphones and tablets. With the wide-deployment of the touch screens, gesture input is mainly limited to the front screen of mobile devices. However, finger input on the touch screen may lead to some cumbersome user experiences. For example, touch screen is not working well when the user's finger is sweaty. Moreover, user's finger often has the occlusion problem [31], *i.e.*, the surface of the finger often blocks the view of some content displayed on the touch screen during the interaction process.

Authors' addresses: Lei Wang, Peking University, China, Email:wang_l@pku.edu.cn; Xiang Zhang, University of New South Wales, Australia and Harvard University, United States; Yuanshuang Jiang; Yong Zhang, Shenzhen Institute of Advanced Technology, CAS, China; Chenren Xu; Ruiyang Gao; Daqing Zhang, School of Electronics Engineering and Computer Science, Peking University, Beijing, China.

2474-9567/2021/6-ART82 $15.00
https://doi.org/10.1145/3463522

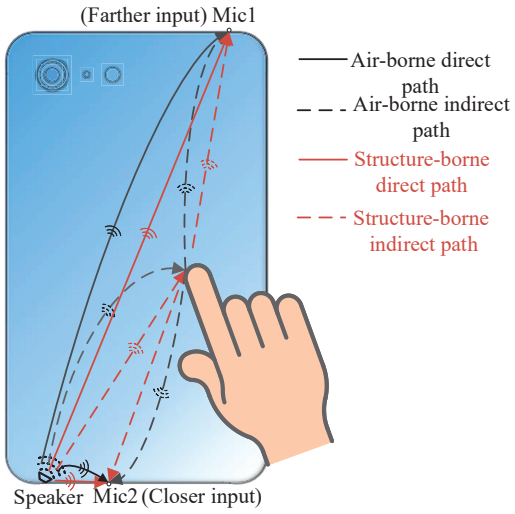


Fig. 1. Ultrasound propagation paths on the back of phone. All paths can be classified into structure-borne paths and air-borne paths. For the sake of simplicity, we called the received signals from the mic1 "the farther input" while the "closer input" for the mic2. The dual received signals, *i.e.*, signals received by two microphones, are used to achieve the accurate sliding gesture recognition based on the structure-borne propagation only.

Back-of-Device (BoD) gestures (*i.e.*, gestures on the back of smartphone) [22, 37] provide a new way of input interface which addresses the above two problems. Additionally, BoD gestures can enrich the user experiences by providing users with the back surface as a writing pad. Along this direction, there has been some considerable work on BoD gestures. Existing works [6, 37] employ the common built-in sensors (*e.g.*, microphone, gyroscope and accelerometer) to sense tap activity. They only focus on this coarse gesture due to the limited information obtained from built-in sensors. Recent study [24] utilizes the sound properties to enable a fine-grained gestures (*e.g.*, swiping, scrolling) sensing system on the surfaces of commodity devices. However, this gesture recognition system relies on the in-air propagation which can be easily interfered by other movements in the air.

As shown in Figure 1, sound signals transmit from the speaker to the microphone through more than one path. All these paths can be classified into structure-borne paths, *i.e.*, sound propagating through the structure of the mobile phone, and air-borne paths, *i.e.*, sound propagating through the air. Beyond the change of air-borne propagation, structure-borne also has a corresponding change when the finger touches the back of mobile phone. Specifically, structure-borne CIR can quantitatively describe the detailed changes relative to the structure-borne propagation.

In this paper, we propose StruGesture, a fine-grained gesture recognition system on the back surface of mobile devices using ultrasonic signals. Our key idea is to leverage the properties of structure-borne CIR to serve as features for gesture recognition. We firstly sketch the structure-borne CIR model, which builds the correlation between the structure-borne propagation and the sliding gestures. Next, we design the whole process from transmitting to receiving signals using built-in speaker and microphones. Then, we focus on extracting structure-borne component from raw CIR measurement and denoise the result by performing differential operation. At last, we propose a novel classification framework based on adversarial deep learning to guarantee the excellent recognition performance.

To handle the gesture recognition on the back of mobile phones, two main challenges must be addressed. The first challenge is to derive the structure-borne component corresponding to the targeted gestures among all propagation. While traditional gesture recognition systems are mainly based on air-borne propagation [12, 24], it

is challenging to extract pure structure-borne component from the hybrid CIR signals. To overcome this challenge, we firstly design a Zadoff-Chu (ZC)-based OFDM signal to get CIR sequence with sub-sample resolution. We expect that the more precise CIR can depict higher resolution of the propagation path information, which enables extracting the structure-borne path component. Secondly, we propose a self-calibration scheme to combat the ambiguity of the structure-borne and air-borne components based on their adjacent properties. In this way, the structure-borne and air-borne components will be moved to the adjacent locations. We then propose a peak selection algorithm based on the amplitude relation and sequential order to determine the air-borne peak and structure-borne peak, respectively. Based on the located structure-borne and air-borne peaks, we obtain the air-borne direct component and the structure-borne component. Through the subsequent signal purification step, the structure-borne component is singled out and the features related to gestures are extracted accordingly.

The second challenge is to eliminate the influence brought by inter-subject difference. Most of the existing studies are focusing on the person dependent situation where the training set and testing set are collected from the same subject. However, the real-world deployment requires person independent setting, in which the training data are collected from a batch of subjects while the testing set from an unseen subject. To address this challenge, we propose an end-to-end unified adversarial learning framework (Section 6) that can capture distinctive gesture-specific representations while alleviating the inter-subject difference.

In summary, the main contributions of this work are highlighted as follows:

- To the best of our knowledge, this is the first structure-borne propagation (SBP)-based sliding gesture sensing research on smartphones demonstrating accuracy and robustness advantage than alternative approaches based on air-borne approaches.
- We propose a ZC-based OFDM de/modulation scheme to derive CIR measurement with much higher resolution and lower time cost than most traditional de/modulation schemes. After deriving the precise CIR measurement, we further put forward novel approaches, including self-calibration and peak detection, to separate the structure-borne component from the hybrid CIR sequence.
- We propose a novel framework for discriminative gesture recognition via purified gesture-specific representations that are learned through adversarial training.
- We design extensive experiments to evaluate our system. The results show that our framework outperforms all the competitive cutting-edge counterparts. Moreover, our system shows appealing performance over 9 usage scenarios, which sheds the light for the practical applications in the future.

The whole paper is organized as follows: we propose the structure-borne propagation model which builds the correlation between the structure-borne CIR and the sliding gestures in Section 3 . Section 4 mainly analyzes the design flow of OFDM signal, which undergoes the modulation and demodulation steps at the transmitter and receiver end. Based on the OFDM signal, Section 5 mainly describes the details to extract the structure-borne component which is highly related with the sliding gestures. Section 6 proposes a novel end-to-end classification framework to accurately recognize the user's gesture based on the input structure-borne signals.

2 RELATED WORK

Existing works on gesture sensing can be divided into three categories: BoD Gesture Sensing, Non-BoD Gesture Sensing and In-air Gesture Sensing.

2.1 BoD Gesture Sensing

Gestures performed on the back of devices are popular ways to enrich user experiences [6, 11, 24, 32, 34, 37]. Buit-in camera can be used to detect the finger movement on the back of mobile phone. LenGesture [34] studies the feasibility of detecting the finger movement just above the rear camera. Back-Mirror [32] uses smartphone with additional mirror, which reflects the back surface to the rear camera, to capture gestures with high resolution

in a large range. However, these schemes either have limitation of sensing range or require additional hardware to extend sensing area. Built-in sensors (*e.g.*, accelerometer, gyroscope, *etc.*) can also be used to sense tap movement [6, 37]. However, the derived information from sensor readings is so limited that only coarse-grained gesture (*i.e.*, tap) can be sensed. A capacity-based prototype is developed in [11], which sense user's touch input. However, this system not only requires hardware modification but also needs an extra capacitive touch controller attached to the user's arm, which is annoying to users. In addition, Vskin [24] performs fine-grained swiping gesture sensing on the back of smartphone. However, the swiping gesture sensing scheme is based on the air-borne propagation, which is easily interfered by nearby noisy movements in the air. For example, the adjacent finger's irrelevant movement will lead to the sharp degradation of interaction accuracy.

2.2 Non-BoD Gesture Sensing

Beyond BoD interaction, several gesture sensing schemes based on different types of surfaces have been proposed [5, 8, 13, 14, 27, 38]. Vibration-based [13, 14] schemes enable tapping location or gestures-based authentication on solid-surface with high accuracy. However, they are deployed on the dedicated device. ForcePhone [27] explores force-sensing capability to the touch screen of mobile phone. However, it uses linear chirp signal to sense force and touch relying on the magnitude change because of structure-borne propagation. It can not capture the fine-grained channel state information through the captured magnitude measurement at a low sampling rate. In contrast, our system use OFDM signals to derive complex signals, which includes the magnitude and phase information of multiple propagation paths. In this way, our system studies fine-grained sliding gesture recognition system rather than squeezing detection scheme[27]. TapSense [8] utilizes the tapping sound to recognize the touch movement on the touch screen. SoundWrite [38] and Ipanel [5] both capture the acoustic signals generated by sliding of fingers on the table for sensing. However, these audible sound based recognition systems are easily influenced by ambient sound noise.

2.3 In-air Gesture Sensing

Sensing in-air gestures has been widely studied in many fields (*e.g.*, Camera signal, Radio Frequency (RF) signal and ultrasound signal).

As for Camera signal, Kollorz *et al.* [10] presents a camera-based approach for gesture classification using x- and y-projections of the image and optional depth features. Wang *et al.* [28] presents a superpixel-based hand gesture recognition system with Kinect depth camera.

As for RF signal, prior works [2, 26, 29, 35] take advantages of the fine-grained Channel State Information (CSI) available from WiFi signals to track or recognize gestures with high accuracy. WiGest uses the changes in WiFi RSSI through three wireless links to recognize a special set of gestures, and achieves a recognition accuracy of 96% [1]. WiDraw applies Angle-Of-Arrival (AOA) measurements to reach the resolution of 5 cm of tracking [25].

As for ultrasound signal, several ultrasound based gesture recognition schemes have been proposed [3, 4, 7, 16, 17, 19, 23, 36]. AudioGest [19], SoundWave [7] and Multiwave [17] utilize Doppler effect to recognize small set of predefined gestures. LLAP [30] use continuous wave to track hand with high accuracy. FingerIO [15] transmits OFDM modulated frames to locate moving finger. Strata [20] tracks phase changes at different time delays corresponding to fine-gained hand locations.

However, gestures in all the aforementioned sensing systems, especially camera-based system, are performed at remote areas away from sensing devices. Furthermore, RF and Ultrasound signal based propagation is vulnerable to other irrelevant movement in the air. In comparison, our system just uses the structure-borne sound signals to sense gestures performed on the back of the mobile devices, which is immune to in-air interference.

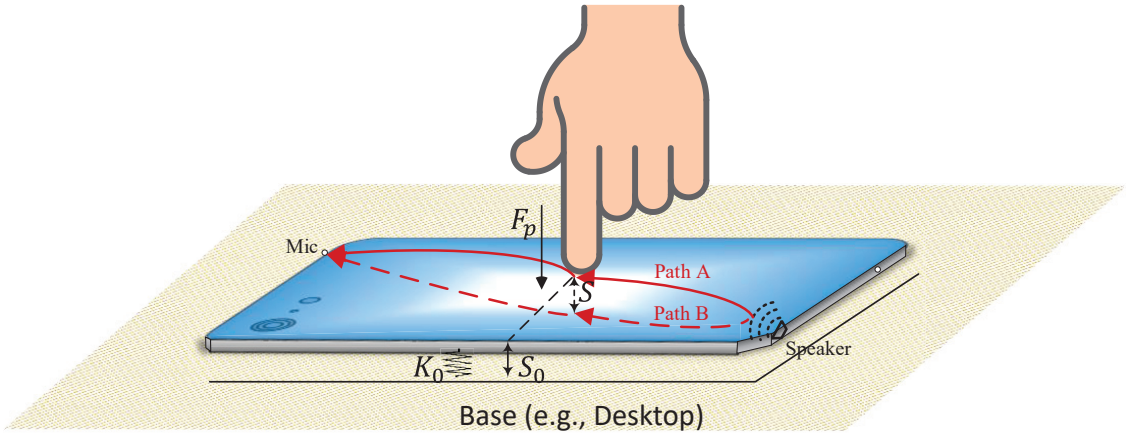


Fig. 2. Structure-borne path's change under the pressure. The vibrating phone caused by sound propagation is considered a spring mass damper system. When a force F_p is applied on the surface, there is a corresponding displacement S. Meanwhile, the structure-borne path will have a relevant change from path A to path B.

3 STRUCTURE-BORNE PROPAGATION MODEL

In this section, we focus on describing the theoretical model of structure-borne propagation regarding the gesture performing on the back surface of mobile phone.

3.1 Model Description

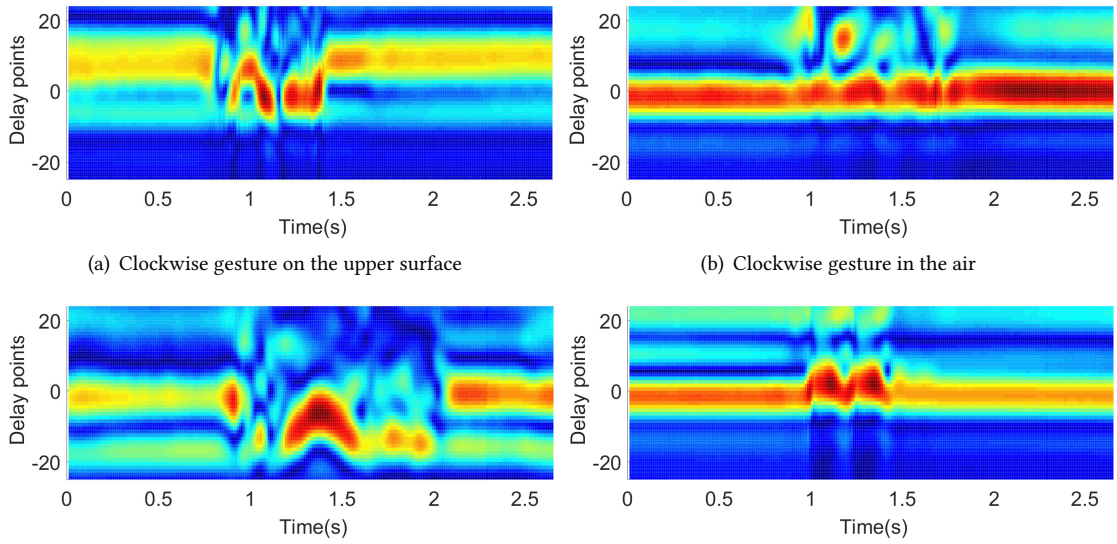
While the user touches the surface of mobile phone as shown in Figure 1, the sound signal will encounter the finger. Then, part of signal will be directly absorbed by the finger and other parts will propagate back to the microphone along with the reflection path [14]. Therefore, the touching location information is related to the reflection path during the whole structure-borne propagation, which demonstrates the high correlation between the structure-borne reflection path change and gestures.

Furthermore, the finger pressure will also affect the structure-borne propagation. We give the Figure 2 to understand this phenomenon. The vibrating phone caused by sound propagation is considered a spring mass damper system [14, 27]. When the sound wave is played at the speaker, the phone can vibrate with stable amplitude S_0 with effective spring coefficient K_0. When there is an external force due to finger pressure, the vibration amplitude will be reduced according to conservation of energy, which can be described as:

$$\frac{1}{2}K_0S_0^2 = \frac{F_p}{S_0 - S}S^2 \tag{1}$$

The vibration displacement S is corresponding to the applied force F_p by the touching finger. Upon that the structure-borne path will have a relevant change (*e.g.*, from path A to path B as shown in Figure 2). Note that the path change is obviously highly correlated with the touching location. Therefore, the ultrasonic signal's structure-borne propagation can be utilized as suitable feature to discriminate different sliding gestures accordingly.

Assume the baseband signal at time t is $s(t)$ and it will be up converted with the carrier frequency f_c followed by transmitting. According to Linear Time Invariant system (LTI) theory, the received baseband signal corresponding to propagation path i can be represented as: $\alpha_i S(t-\tau_i-\tau_0)e^{-j2\pi f(\tau_i+\tau_0)}$, where τ_i and α_i are the corresponding time delay determined by distance of the travel path and amplitude representing the signal attenuation, respectively. τ_0 denotes the initial time delay due to the hardware imperfectness, which can be mainly divided into two parts: the delay from the audio file to the playing sound at the speaker side and the delay from the recording sound to the audio file at the microphone side. Note that this delay is constant when the system is always working while



(a) Clockwise gesture on the upper surface

(b) Clockwise gesture in the air

(c) Clockwise gesture on the bottom surface

(d) Double-click gesture on the surface

Fig. 3. CIR patterns under different scenarios

random for the new restarts. Since the sound wave is a longitudinal wave[21] which can propagate through air and structure, all propagations can be further grouped into air-borne and structure-borne propagation as shown in Figure 1. The received baseband signal can be represented as:

$$\begin{aligned} R(t) &= \sum_{m=1}^{L_a} \alpha_m S(t-\tau_m-\tau_0) e^{-j2\pi f_c(\tau_m+\tau_0)} + \sum_{n=1}^{L_s} \alpha_n S(t-\tau_n-\tau_0) e^{-j2\pi f_c(\tau_n+\tau_0)} \\ &= S(t) \circledast (H_a(t) + H_s(t)) \end{aligned} \tag{2}$$

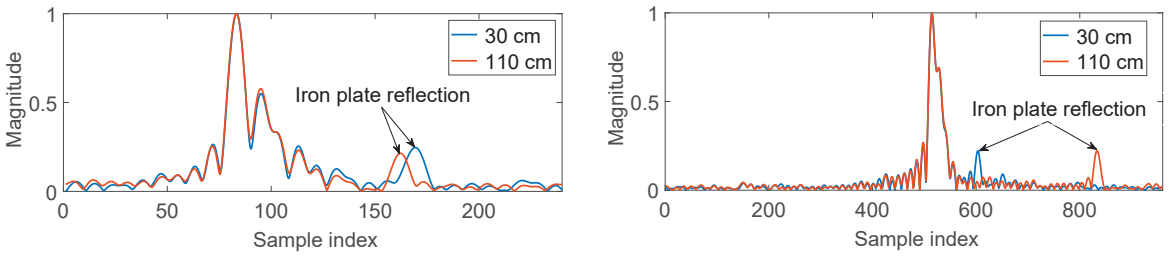
where $\circledast$ denotes the convolution operation and $H_s(t)$ is the CIR relative to structure-borne propagation. Hence,

$$H_s(t) = \sum_{n=1}^{L_s} \alpha_n \delta(t-\tau_n-\tau_0) e^{-j2\pi f(\tau_n+\tau_0)} \tag{3}$$

where $\delta(t)$ is Dirac's delta function. This indicates that the structure-borne CIR can quantitatively describe the characteristics of structure-borne propagation including time delay and amplitude.

3.2 Model Verification

Since the structure-borne and air-borne direct path are both the shortest paths, the sound energy corresponding to these two paths are strongest. Additionally, since the sound speed in solid is 100× faster than in air, we can easily locate the structure path along the CIR time delays. Figure 3 shows the amplitude of CIR estimations along the timeline when the finger is sliding on the back surface and in the air, respectively. For the vertical axis, the delay point denotes the path length from the speaker to the microphone. The hot region indicates a peak at the corresponding path length in the CIR estimation. For convenience, we define the delay point of 0 to be the structure-borne path. We then conduct four representative experiments to verify the structure-borne propagation model. We firstly use the finger to slide on the upper surface of the phone with the clockwise gestures. From Figure 3(a), we observe that CIR amplitude is changing during sliding period ($0.7 \sim 1.4s$) and delay interval ($-10 \sim 10$). Note that the delay interval ($-10 \sim 10$) is mainly dominated by the structure-borne propagation. Compared with the sliding movement on the surface, the same gestures in the air (*i.e.*, 1 cm above the surface)



(a) CIR profile with frame size of 240

(b) CIR profile with frame size of 960

Fig. 4. CIR profile of the closer input with different frame sizes

has a negligible effect on the structure-borne component, as shown in Figure3(b). To demonstrate the stability of the pattern, we slide on the bottom surface with the same gesture. From Figure3(c), we observed that the pattern during the sliding period $0.9 \sim 1.9s$ is similar to that in Figure 3(a) since they both have a similar arc-shaped pattern for the hottest region. To study the diverse pattern corresponding to various gestures, we perform the double-click gesture shown in Figure3(d), which implies the uniqueness of the specified gesture.

4 ZC-BASED OFDM DESIGN

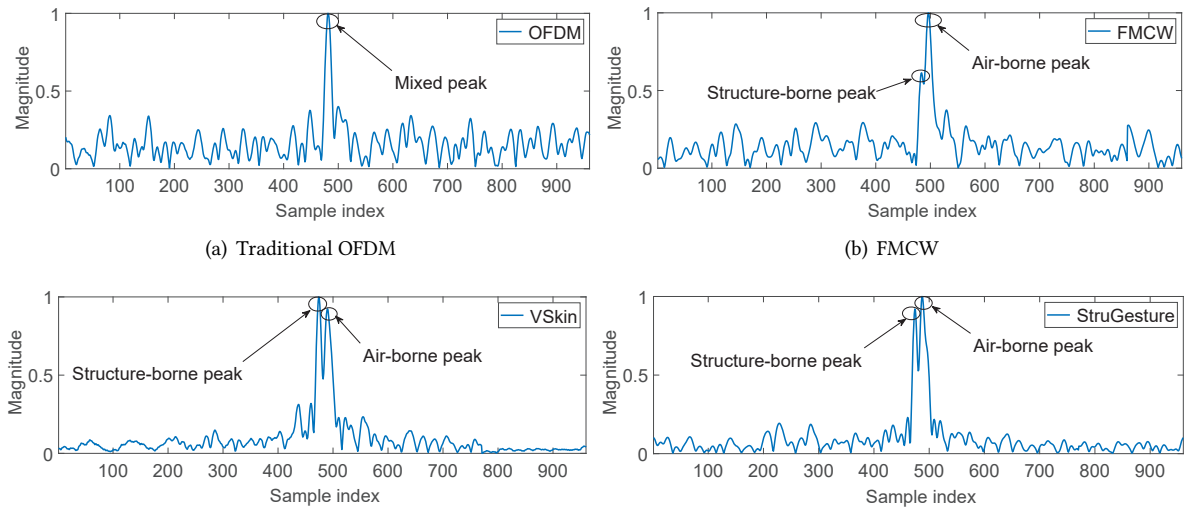
To capture the fine-grained channel information, we need to derive not only the amplitude change but also the phase change. This is because the only amplitude between the adjacent samples may have a little change so that it is difficult to capture the fine-grained movement with the amplitude information, such as ForcePhone [27]. As we know, traditional OFDM system provides us the channel information with the amplitude and phase [15]. However, it is still not sufficient to achieve fine-grained sensing due to the randomness of the transmitted sequence. Additionally, the time cost is high due to series of complicated operations, for example, up-conversion, down-conversion, and low-pass filter. In this section, we focus on designing a ZC-based OFDM system to address thse two issues.

4.1 Parameter Analysis

Our system utilizes the transmission of frequency narrowly ranging in $17 - 23\ kHZ$ (*i.e.*, $B = 6\ kHz$) with the central frequency $20\ kHz$ and sampling rate $48\ kHz$, which is a inaudible to most people [18]. The transmission consists of consecutively repeated frames. When the microphone receives the signal, it derives the time delay through the circular shift.

The selection of frame size, N, should satisfy the following requirements from both the technical and engineering aspects. For the aspect of technology, the candidate frame size should be designed to reduce the surrounding interference as much as possible. This is because the larger frame size can reduce the disturbance of echoes from larger distance according to the time shifting theorem. To demonstrate this, we give Figure 4 to show the difference in CIR profile for the frame size $N = 240$ and $N = 960$, respectively. We first put the mobile phone on an empty desktop. We then move an iron plate from 30 cm to 110 cm away from the mobile phone. For the frame size of $N = 240$, the reflected signals corresponding to 30 cm and 110 cm have considerable overlap, as shown in Figure 4(a). This is because the sensing range for $N = 240$ is within $c \cdot N/(2 \cdot F_s) = 85.8\ cm$. The reflected signals outside this range will be superimposed on the target signals. In contrast, the frame size of $N = 960$ implies that the sensing range is within 343 cm and the same previous reflection signals can be well separated, as shown in Figure 4(b). Meanwhile, the weak reflected signals beyond this range is negligible due to energy attenuation. Overall, we choose $N = 960$, which is sufficient to meet the requirements of our system.

For the aspect of engineering, the candidate frame size should be firstly integer multiples of the period of the transmitted passband cos/sin signals (*i.e.*, $rep(\frac{N \cdot f_c}{F_s}, 1) = 0$) so that both the transmitted and received signals can be viewed as the continuous periodical signals. The amount of timing delay corresponding to the static target



(a) Traditional OFDM (b) FMCW

(c) VSkin (d) StruGesture

Fig. 5. Temporary CIR profile comparison via typical demodulation schemes for the farther input signals

	Traditional-OFDM	**FMCW**	**VSkin**	**StruGesture**
Sumsang S7	7.01 *ms*	8.29 *ms*	15.65 *ms*	3.56 *ms*
Nexsus 6	8.42 *ms*	9.23 *ms*	15.91 *ms*	4.35 *ms*

Table 1. Time consumption comparison between typical demodulation schemes

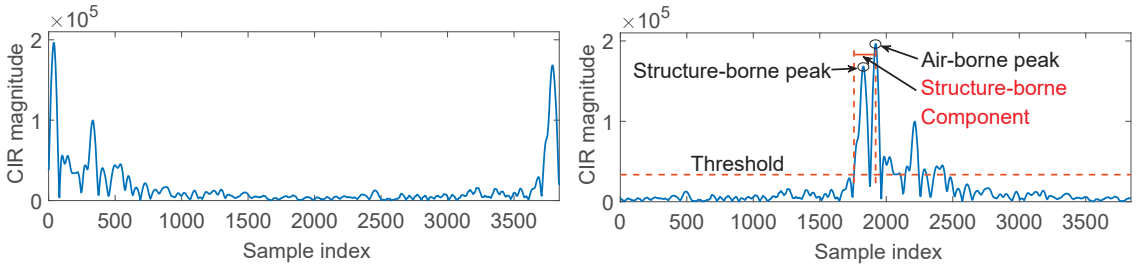
can be, therefore, regarded as constant. Secondly, to enable StruGesture on most mobile devices, the frame size should be the combination of radix-3, radix-5, radix-2 so that Inverse Fast Fourier Transform (IFFT) and Fast Fourier Transform (FFT) can be calculated.

4.2 Modulation Process

For the modulation process, we firstly choose temporal sequence with the size of $N_c = \frac{B \cdot N}{F_s} + 1$ to satisfy the bandwidth of modulated signal. For example, with the sampling rate $F_s = 48\ kHz$ and frame size of $N = 960$, we have $N_c = 81$. Furthermore, the temporal sequence with good auto-correlation property is provably optimal to derive the CIR with high resolution. We therefore choose ZC sequence, which meets the requirement [24], as the temporal sequence. Secondly, we transform the temporal sequence into frequency sequence $X[n]$ by performing N_c-point FFT operation. Thirdly, we rearrange the frequency points to $X_s[n]$ by performing FFT shift so that the DC component is at center of the sequence. Fourthly, we copy the rearranged sequence to the positive frequency part of $\widehat{X}[n]$ with length N. We then copy the conjugate of $X_s[n]$ to the negative part of $\widehat{X}[n]$. The rest part of $\widehat{X}[n]$ is padded with zero. Finally, we perform N-point IFFT on the zero-padding complex valued sequence $\widehat{X}[n]$ to derive the real-valued transmitted sequence $x[n]$. Since each frame fully contains multiple periods of the passband signals, the transmitting sequence can be generated and played on most mobile devices.

4.3 Demodulation Process

At the receiver side, we derive the CIR by leveraging frequency domain multiplication instead of traditional de/modulation schemes, such as [15, 24, 27], to significantly reduce the time cost. We firstly segment the received sequences into the frames of length N. Secondly, we perform an N-point FFT on each frame to derive the frequency domain representation $Y[n]$. Thirdly, we extract the passband frequency components corresponding to $X_s[n]$ and



(a) Raw CIR profile (b) Calibrated CIR profile

Fig. 6. CIR profile before and after self-calibration

divide it by the transmitted frequency sequence $X_s[n]$ to get the Channel Frequency Response (CFR) (*i.e.*, $H[n]$). We rearrange the frequency domain sequence and perform zero-padding to expand the frequency bandwith length. The length of sequence is expanded from N to N_e and CFR is transformed to $\widehat{H}[n]$, $n = 0, \ldots N_e - 1$. We set $N_e = 4N$, which means the resolution is four times than that of the raw sequence. Finally, we perform N_e-point IFFT on the zero-padding sequence to get the CIR at time t as follows:

$$h_t[n] = \sum_{i=1}^{L_s+L_a} sinc(n - k_i) e^{-j2\pi k_i f_c N/(N_e f_s)}, n = 0, \ldots N_e - 1 \tag{4}$$

According to time-frequency characteristic [9], the good auto-correlation of ZC sequence gives an ideal CIR with the limited bandwith.

4.4 The Novelty and Advantages

This section's key novelty is on proposing a ZC-based OFDM scheme to separate the structure-borne and air-borne components. The main advantages of our ZC-based OFDM scheme are from two aspects. Firstly, the ZC-based OFDM scheme has a much higher resolution than traditional de/modulation schemes, *e.g.*, traditional OFDM [15] and FMCW [27]. We use a mobile phone on the empty desk to explore the resolution of the four de/modulation schemes above. As shown in Figure 5(a), the traditional OFDM scheme is insufficient to separate the structure-borne and air-borne peaks based on the ordinary sequence. There is only one mixed peak along with the temporary CIR profile, and the values of other points are relatively large due to the poor auto-correlation. FMCW signals are similar to OFDM signals even though the two peak points are slightly separated, as shown in Figure 5(b). In contrast, these peaks can be well distinguished with ZC-based OFDM and VSkin signals [24] due to the good auto-correlation property, as shown in Figure 5(c) and Figure 5(d). Additionally, the values of other points, which indicate no propagation, are far smaller than OFDM and FMCW signals.

Secondly, ZC-based OFDM scheme has a much less computational complexity than the aforementioned de/modulation schemes, as shown in Table 1. Instead of using the time domain down-conversion as illustrated in [15, 24, 27], we leverage the only frequency-domain shift to perform the time-domain down-conversion, which can significantly reduce the computational cost. Additionally, while VSkin uses the time-domain correlation to get CIR, we only use the frequency-domain multiplication to ensure the equal resolution and further reduce the time complexity.

5 STRUCTURE-BORNE COMPONENT MEASUREMENT

5.1 System Self-calibration

As shown in Figure 1, among all the multi-path propagation signals, there are air-borne signals (signals propagated in the air) and structure-borne signals (signals propagated through the solid). These two types of

ALGORITHM 1: Peak Selection

Input: Calibrated CIR sequence $\widehat{h_t}[k]$
Output: Selected peak points
Locate all the peaks (local maximum point);
Sort the peaks by their amplitude $|\widehat{h_t}[k_i]|$, $i = 1, \ldots, N_p$:
$|\widehat{h_t}[k_1]| > , \ldots, > |\widehat{h_t}[k_{N_p}]|$;
Initialization: $Candidate1 \Leftarrow k_1$, $Candidate2 \Leftarrow k_1$;
for $i = 2, \ldots, N_p$ **do**
    if $|k_i - k_1| < k_{thrd}$ *and* $(|h_t[k_i]|) > h_{thrd}$ **then**
        $Candidate2 \Leftarrow k_i$;
        break;
    end
end
Structure-borne peak $\Leftarrow \min[Candidate1, Candidate2]$;
Air-borne peak $\Leftarrow \max[Candidate1, Candidate2]$;

signals are both further divided into: direct signals (signals propagated from the speaker to microphones through LOS path) and indirect path (signals reflected by the object in the environment). As mentioned before, sound speed in the solid is 100× faster than that in the air, which implies that the direct and indirect signals corresponding to the structure-borne propagation will be mixed due to limited resolution. Conversely, they can be captured and differentiated in the air. As we know, the direct path has the shortest path length among all air-borne propagation, which indicates the lowest energy decrement and hence corresponds to a prominent peak in the CIR profile. Similarly, the structure-borne propagation also has a mixed peak over the CIR profile theoretically. For the sake of description, we use **air-borne peak** and **structure-borne peak** to denote the peak of air-borne direct propagation and structure-borne propagation, respectively.

Through investigation, we find that most off-the-shelf mobile phones (Iphone series, Samsung series, *etc.*) have at least one built-in speaker and two built-in microphones similar to the deployment in Figure 1, of which one microphone is much closer to the speaker than the other. Our system is aimed at sensing the sliding gestures by utilizing the stereo input (the closer input and the farther input).

For the closer input, due to the short distance between the speaker and the microphone, the air-borne peak and the structure-borne peak will be mixed into the only peak, as shown in Figure 4. Therefore, it is easy to determine the location of the mixed peak directly by searching the maximal peak.

For the farther input, the structure-borne and air-borne direct propagation are separate because of the large propagation delay difference. However, we cannot directly identify the structure-borne and air-borne direct peak due to the random initial propagation delay, as shown in Figure 6(a). To combat the ambiguity, we perform a calibration scheme based on the property of adjacency with these two components. We firstly locate the maximum peak point over the CIR profile. The selected peak point must corresponds to either structure-borne or air-borne direct components. Once one of components is determined, the other is around it theoretically. Inspired by this, we then shift the CIR sequence circularly until the maximum peak is at the central position as shown in Figure 6(b). The calibrated CIR is given by:

$$\widehat{h_t}[k] \quad = \quad h_t[(k - (N_e/2 - P)) \mod N_e], k = 0, \ldots N_e - 1 \tag{5}$$

where $P = \underset{k}{\text{argmax}}(h_t[k])$ and notation k represents the k-th delay point value relative to the calibrated CIR. Note that although the time delay varies with the circular shifting, the relative relation between paths is retained.

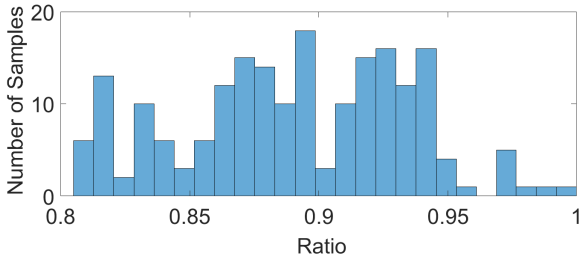


Fig. 7. The distribution of 200 samples with the ratio of the 2nd peak to the 1st peak

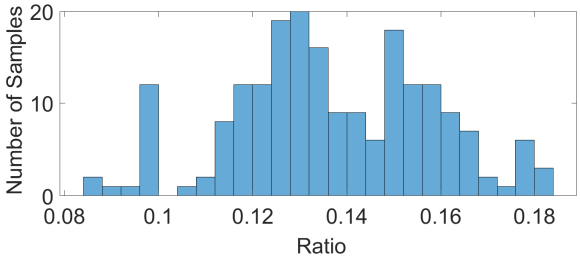


Fig. 8. The distribution of 200 samples with the ratio of the maximal noisy peak to the structure-borne peak

5.2 Structure-borne Component Identification

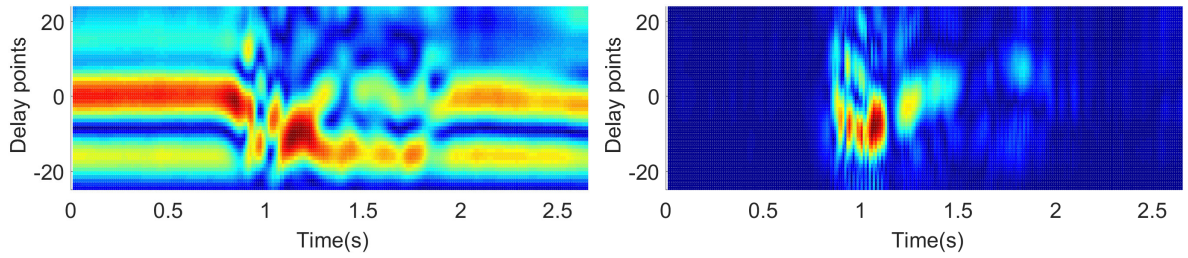
To identify the structure-borne component, the first step is to use Algorithm 1 to select the peak points corresponding to the structure-borne propagation and air-borne direct propagation. To select the exact peak points, we firstly locate all the peaks from the calibrated CIR sequence $h_t[k]$, which are the points with local maximum amplitudes. Subsequently, we sort peak points by their amplitude as the labels, *i.e.*, $i = 1, \ldots, N_p$, as shown in Figure 6(b). Therefore, we get:

$$|\widehat{h}_t[k_1]| > , \ldots , > |\widehat{h}_t[k_{N_p}]| \tag{6}$$

We then initialize the highest peak as the $candicate1$ and $candidate2$. Starting from the second highest peak, we take out current peaks one-by-one in the descending order of amplitudes. If the interval between the current peak with the highest peak is within k_{thrd} and the current peak's amplitude is larger than h_{thrd}, we treat this peak as the $candidate2$, and other peaks are removed meanwhile. We set the threshold k_{thrd} to be 112 and h_{thrd} to be $0.8 \times |\widehat{h}_t[k_1]|$, where 112 implies that the largest distance between the speaker and microphone is within 20 cm which satisfies most devices. Meanwhile, since the ratio of the structure-borne peak to the air-borne peak is greater than 0.8, as shown in Figure 7, $0.8 \times |\widehat{h}_t[k_1]|$ ensures that most noisy peaks are removed. Note that if the speaker and microphone are very close to each other, $candicate1$ is equal to $candidate2$ as illustrated in Section 5.1. As we know, the received structure-borne signals are earlier than the air-borne signals because of the higher speed. We thus get the structure-borne peak as $\min[Candidate1, Candidate2]$ and air-borne peak as $\max[Candidate1, Candidate2]$.

Based on the located structure-borne and air-borne peaks, the interval of $|Candidate1 - Candidate2|$ is filled with the structure-borne component and air-borne direct component. Note that the air-borne direct component is stable and entirely unrelated to the movement, it will be eliminated in the following step. Additionally, we also select interval before the structure-borne peak, of which the amplitude is larger than empirical threshold $0.2 \times |\widehat{h}_t[\min[Candidate1, Candidate2]]|$, to include the majority of the structure-borne component. As shown in Figure 8, since the ratio of the highest peak, which is among all noisy peaks before the structure-borne, to the structure-borne peak is distributed in the range of $[0.08, 0.18]$, $0.2 \times |\widehat{h}_t[\min[Candidate1, Candidate2]]|$ can ensure that the noisy peaks before the structure-borne peak are all removed.

As shown in Figure 6(b), static components (including air-borne direct component and some structure-borne static components) dominate the raw CIR, which result in difficulty in highlighting the signals relative with sliding movements. We reduce the effect of static components by performing differential operation on complex-valued CIR sequence instead of intuitive CIR amplitude. This is because the amplitude has a small change within a short duration, the direct differential operation will remove most movement-related signals as well as the static signals. In contrast, the difference of complex valued CIR retains the movement-related signals since the phase has a obvious change with the same path length change. We give Figure 9 to illustrate the resulting performance of



(a) Amplitude of raw CIR

(b) Amplitude of differential CIR

Fig. 9. CIR comparison

signal purification. Compared with the amplitude of the raw CIR in Figure 9(a), the amplitude of differential CIR is concentrated during the sliding period, *i.e.*, 0.85 ∼ 1.85 *s*, as shown in Figure 9(b).

5.3 Sliding Detection and Segmentation

We use the property of structure-borne CIR regarding one microphone to detect the starting and ending time of the movement. Note that we just consider the scenario with only one microphone because of the high synchronization of two built-in microphones. Furthermore, air-borne CIR is mainly relative to the propagation in the air as shown in Figure 3(b). This means irrelevant movements in the air can also trigger our system if the detecting scheme is based on air-borne propagation. We thus attempt to detect the sliding movement by analyzing the temporal CIR profile of certain structure-borne path in Figure 10(a). Following the differential operation, we derive the subsequent result in Figure 10(b). We then apply a 6-th low pass Butterworth filter on the raw differential CIR sequence to smooth the profile, in which cut-off frequency is set to 10 Hz. This method can effectively reduce detection failure because of short-term pause and high-frequency noises during the sliding movement period. Next, we use a empirical threshold to determine the sliding moment or the silent moment. Figure 10(b) shows that the sliding duration can be detected timely with high robustness. Based on the detection, we regard the signal between the calculated starting and ending time as the final sliding segment.

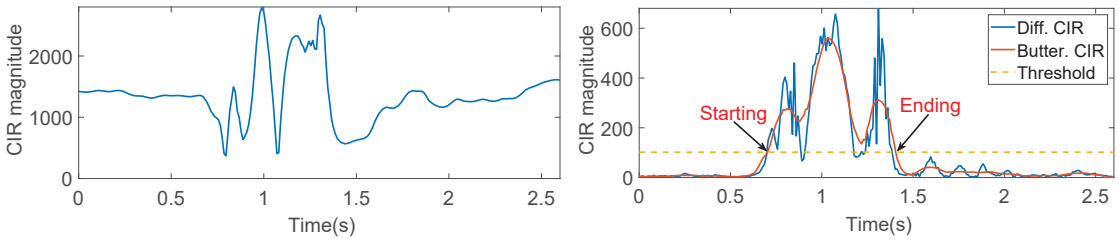
5.4 Normalization

The normalization takes the segmented differential CIR signals and undergoes two steps. As we know, force strength applied on the surface during the sliding period is different between different sliding times even for the same person. In order to reduce the variation of force strength, we firstly normalize the segmented differential CIR amplitudes, which are realized by dividing by the maximum CIR amplitude over each segmented duration. In this way, we can reduce the force strength with different time and retain the diversity corresponding to varied delays simultaneously.

Second, we normalize the segmented duration so that all signals have the same length. In order to remain the intrinsic variation of each segment, we just pad zeros at the end. In this way, we pad all segments into the same length.

5.5 The Novelty and Advantages

This section's key novelty is to identify the structure-borne and air-borne components for each independent microphone based on the systematic approaches, including self-calibration and peak detection. In comparison, previous work [24] proposes to locate the structure-borne peak and air-borne peaks for the farther microphone with the help of the closer microphone. Additionally, this system requires that one of the microphones must be very close to the speaker, which limits its generalization significantly.



(a) Structure-borne CIR profile (b) Differential CIR profile

Fig. 10. Temporary CIR profile before and after differential operation

6 GESTURE RECOGNITION

We propose a novel end-to-end classification framework to accurately recognize the user's gesture based on the input structure-borne signals. In this section, we firstly present the overview of our approach along with the motivation, then report each component in detail.

6.1 Classifier Overview

One major challenge preventing precise gesture classification is the signal corruption. In specific, the acoustic signals contains not only gesture-specific information but also user-specific information. We attempt to extract gesture-specific information while eliminating the user-specific information for more robust and effective gesture recognition. The fundamental idea of this model is to decompose the acoustic data into two parts: one part is rich of gesture patterns but insensitive to the user while another part includes user-related information but insensitive to the gesture. Our target is to extract the gesture-specific representations by purifying the acoustic data. Then, the extracted gesture-specific representations are used to train a powerful classifier for downstream recognition.

We propose a deep adversarial learning algorithm in order to learn pure gesture-specific representations (as shown in Figure 11), which is composed of three crucial components: signal decomposition, user-specific classifier, and gesture-specific classifier.

Firstly, the input signals are decomposed into the user and gesture representations corresponding to the user and gesture information, respectively. Then, the two latent representations are used to reconstruct the input acoustic signals. If the reconstructed signals are equal to the input signals, we regard the learned user and gesture representation as effective features of the decomposed structure-borne signal.

Meanwhile, we design a gesture-specific classifier following the gesture representation for purification. In detail, the classifier aims to recognize the gesture based on the learned gesture representation. If the classifier could achieve competitive performance (*i.e.*, has a low recognition error), we believe that the learned gesture representation contains abundant of gesture-related information. Similarly, a user-specific classifier is designed to purify the user representation. Please note, in the testing stage, the input signals will only be processed by the well-trained signal decomposition and gesture-specific classifier to recognize the gesture. In other words, the user-specific classifier is not used in testing.

6.2 Signal Decomposition

This component receives the structure-borne signals and decompose them into gesture-specific representation and user-specific representation. We denote the input signals by $\mathcal{S} = \{\boldsymbol{S_i} \in \mathbb{R}^{M \times N}\}$. A certain sample $\boldsymbol{S_i}$ has M detailed coefficients and N time delays. Then we have:

$$\mathcal{S} = \mathcal{U} + \mathcal{G} \tag{7}$$

where $\mathcal{U} = \{\boldsymbol{U_i} \in \mathbb{R}^{M \times N}\}$ denotes the set of user-specific component while $\mathcal{G} = \{\boldsymbol{G_i} \in \mathbb{R}^{M \times N}\}$ denotes the set of gesture-specific component.

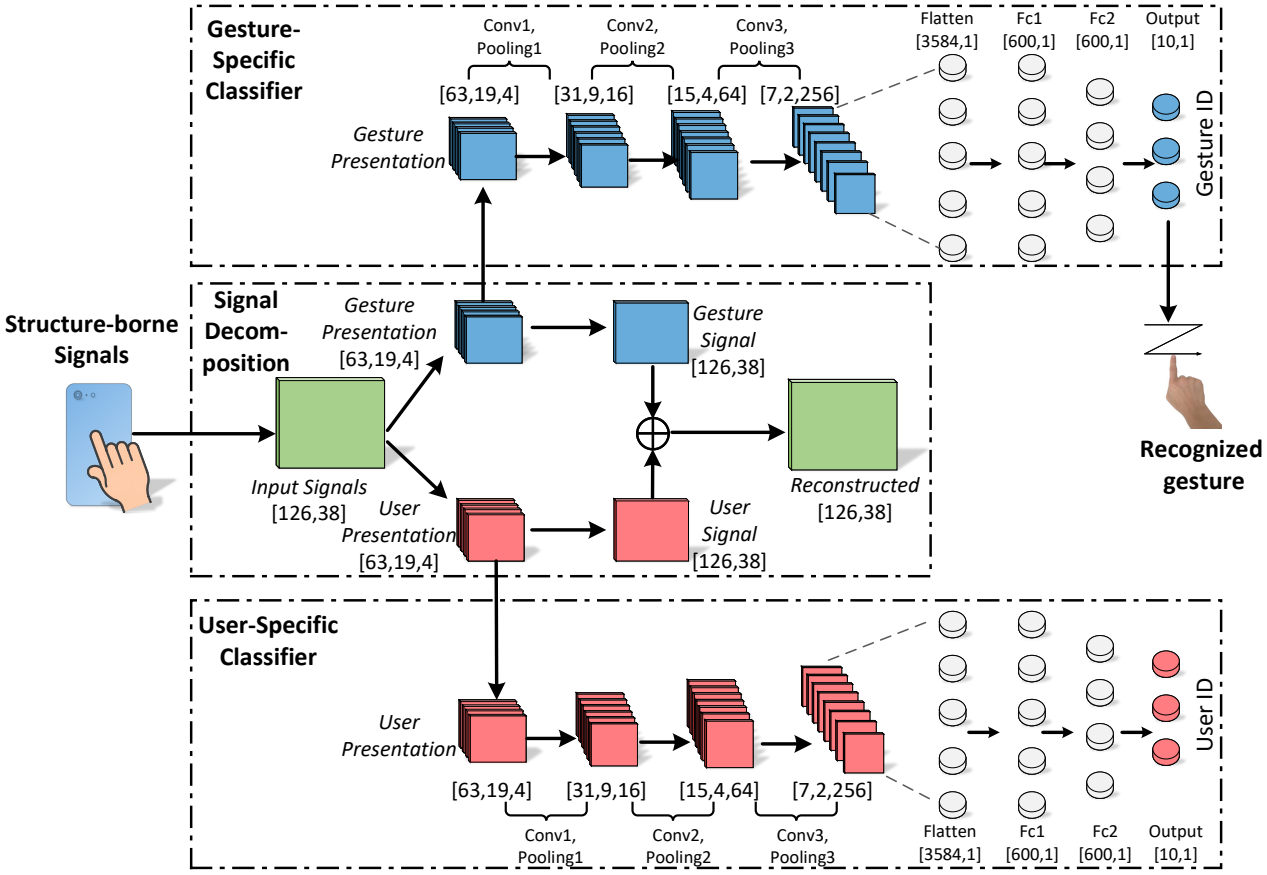


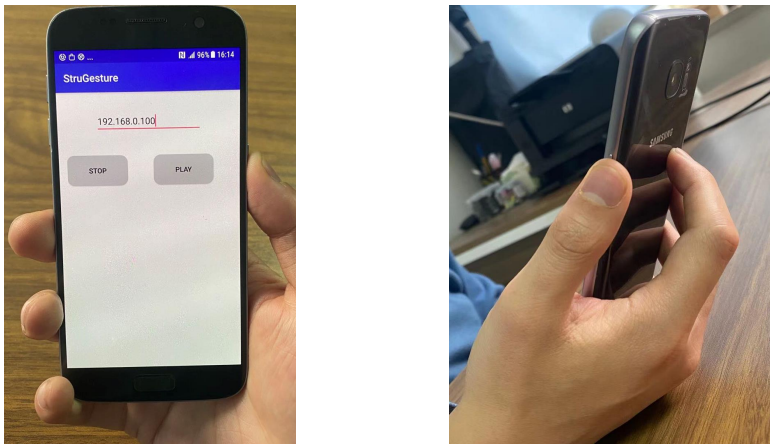
Fig. 11. The proposed gesture recognition framework. In the training stage, the received structure-borne signals are firstly fed into the signal decomposition component in order to extract the gesture and user presentation. The latent representations are sent to the gesture-specific and user-specific classifiers simultaneously. In the testing stage, the input signals will only be processed by the well-trained signal decomposition and gesture-specific classifier to recognize the gesture.

However, in the signal decomposition procedure, the $\mathcal{U}$ and $\mathcal{G}$ cannot be directly calculated through traditional methods. Thus, we propose a novel deep learning-based decomposition model via adversarial learning. Although the input signals have temporal dependencies, our preliminary results show that convolutional layers achieve similar performance with recurrent layers (CNN: 92.6%; RNN: 91.7%) but cost much shorter training time (CNN: 18 s; RNN: 160 s per epoch). A potential reason is that our data has a large number of time steps (126 time slices), which will require massive computing resources if using RNN. Thus, we decide to use convolutional layers to perform signal decomposition. The $\boldsymbol{S}$ is transformed into a latent space through convolutional operation followed by a max-pooling layer[1]:

$$\bar{\boldsymbol{G}} = ReLU(\boldsymbol{w_g} \circledast \boldsymbol{S} + \boldsymbol{b_g}), \tag{8}$$

$$\bar{\boldsymbol{U}} = ReLU(\boldsymbol{w_u} \circledast \boldsymbol{S} + \boldsymbol{b_u}) \tag{9}$$

[1]For simplify, we ignore the sub script.



(a) Front side (b) Back side

Fig. 12. Experimental setup

where $\bar{G} \in \mathbb{R}^{J\times K\times H}$ and $\bar{U} \in \mathbb{R}^{J\times K\times H}$ denote the latent gesture and user representation, respectively. The H denotes the number of convolutional filters. The $\boldsymbol{w_g}, \boldsymbol{b_g}, \boldsymbol{w_u}, \boldsymbol{b_u}$ represent trainable weights and biases.

Then, in order to reconstruct $\boldsymbol{S}$ based on the learned latent representations, we have deconvolutional operations:

$$\boldsymbol{G} = ReLU(\boldsymbol{w'_g} \circledast \bar{\boldsymbol{G}} + \boldsymbol{b'_g}), \tag{10}$$

$$\boldsymbol{U} = ReLU(\boldsymbol{w'_u} \circledast \bar{\boldsymbol{U}} + \boldsymbol{b'_u}) \tag{11}$$

where $\boldsymbol{G}, \boldsymbol{U} \in \mathbb{R}^{M\times N}$ represents the decomposed gesture and user related signals, which have the same shape with the input acoustic signal $\boldsymbol{S}$. The reconstructed signal $\boldsymbol{S'} \in \mathbb{R}^{M\times N}$:

$$\boldsymbol{S'} = (\boldsymbol{G} + \boldsymbol{U})/2 \tag{12}$$

To guarantee the decomposition performance and separate the useful gesture-related information while eliminating the inter-user noise, we force the reconstructed signal $\boldsymbol{S'}$ to approximate the original signal $\boldsymbol{S}$ by minimizing the Euclidean distance between them. In specific, the reconstruction loss function of signal decomposition component is measured:

$$\mathcal{L}_{SD} = \left\| \boldsymbol{S} - \boldsymbol{S'} \right\|_2 \tag{13}$$

6.3 Gesture- and User-specific Classifiers

The architectures of the gesture-specific and user-specific classifiers are highly similar. Here, we take the former as an example for detailed explanation. Convolutional Neural Network (CNN) has been widely used in research areas such as computer vision and natural language process for the excellent spatial feature learning ability. We design a CNN to automatically extract the distinctive information from the received gesture representation $\bar{G}$ which has shape $[J, K, H]$. The output of the classifier is the gesture label of input acoustic signal. As shown in Figure 11, the gesture-specific classifier contains three convolutional layers where each followed by a max-pooling layer, a flatten layer, two fully-connected layers (FC) and an output layer.

The core data flow of the classifier can be formulated as:

$$\boldsymbol{C^i} = ReLU(\boldsymbol{w_g^i} \circledast \boldsymbol{P^{i-1}} + \boldsymbol{b_g^i}), \quad i \in \{1, 2, 3\} \tag{14}$$

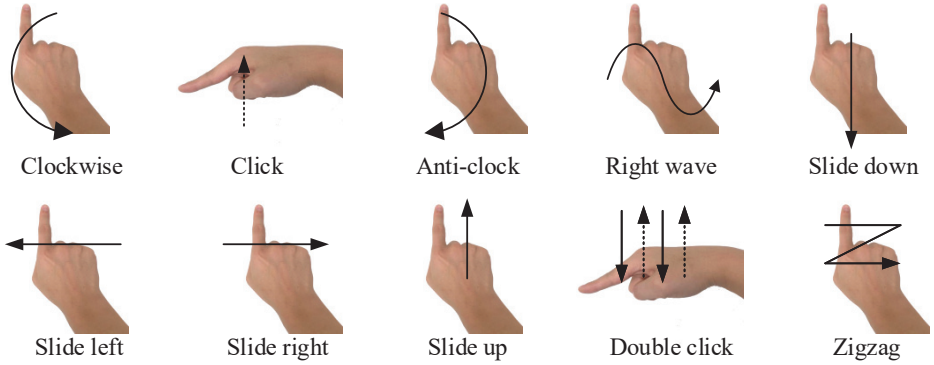


Fig. 13. Ten gestures used in our system. All volunteers are asked to perform these 10 gestures and repeat 100 times with index finger. Note that these figures are only the schematic plot. The experimental setup will be introduced in Section 7.1.

where C^i and P^i denote the i-th convolutional and pooling layer, respectively. If $i = 1$, $P^0 = \bar{G}$; otherwise, the pooling layer can be measured by:

$$P_u^i = \max_{u \in \mathcal{U}} \{C_u^i\}, \quad i \in \{1, 2, 3\} \tag{15}$$

where $\mathcal{U}$ represents the max-pooling perception field and u denotes the u-th element.

The third pooling layer P^3 is flattened to 1-D vector F^0 and then fed into the fully-connected layers:

$$F^i = sigmoid(w_f^i F^{i-1} + b_f^i), \quad f \in \{1, 2, 3\} \tag{16}$$

where *sigmoid* denotes the activation function and F^i denotes the i-th FC layer. The learned F^3 represents the probability of each gesture category.

The classifier receives the learned $\bar{G}$ and produces the predicted gesture $\hat{y_g}$ (i.e., $\hat{y_g} = F^3$). We use cross-entropy function to measure the loss:

$$\mathcal{L}_g = -\sum_{c=1}^{C} (y_{g,c} log(\hat{y_{g,c}})) \tag{17}$$

where y_g and C denote the ground truth of gesture and the overall category of gestures, respectively.

Similarly, we can calculate the loss function of the user-specific classifier $\mathcal{L}_u$:

$$\mathcal{L}_u = -\sum_{c=1}^{C} (y_{u,c} log(\hat{y_{u,c}})) \tag{18}$$

where y_u and $\hat{y_u}$ denote the ground truth and predicted labels of user, respectively.

6.4 Training Details

In the proposed approach, we have three loss functions: the reconstruction loss $\mathcal{L}_{SD}$ in signal decomposition, the classification loss $\mathcal{L}_g$ in gesture-specific classifier, the classification loss $\mathcal{L}_u$ in user-specific classifier. The reconstruction loss $\mathcal{L}_{SD}$ is expected to guarantee the combination of the gesture- and user-representations (i.e., $\bar{G}$ and $\bar{U}$) can approximate the input acoustic signal S, but does not concern whether the decomposed representations informative or not. In contrast, the classification loss $\mathcal{L}_g$ and $\mathcal{L}_u$ target the informative of the decomposed $\bar{G}$ and $\bar{U}$ but don't care whether they can reconstruct S perfectly.

Thus, we propose an adversarial training strategy to jointly train both two counterparts:

$$\mathcal{L} = \alpha \mathcal{L}_{SD} + \beta(\mathcal{L}_g + \mathcal{L}_u) \tag{19}$$

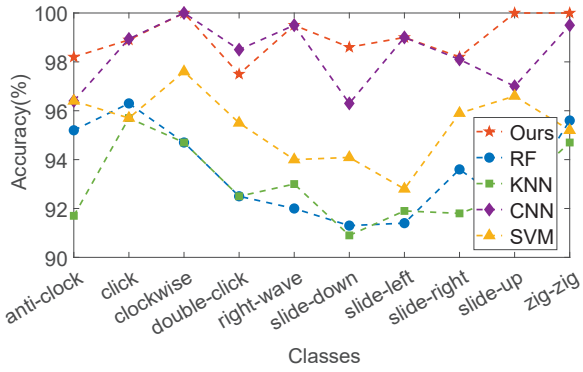


Fig. 14. Accuracy for different algorithms under PD

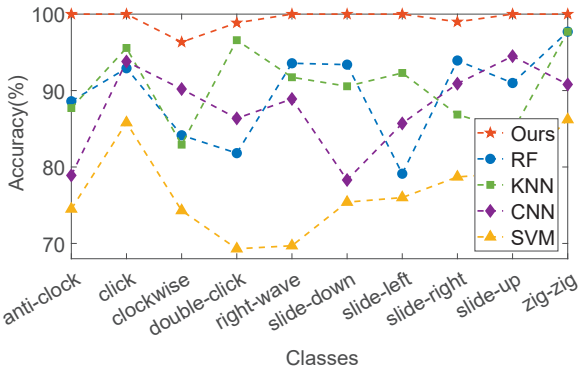


Fig. 15. Accuracy for different algorithms under PSD

where α and β and trainable weights in order to adjust the importance of signal decomposition component and the user/gesture-specific classifier. The α and β are both initialized as 0.5 and will be automatically adjusted and converged to trade-off point based on back-propagation of the loss function. The above equation can make sure that the two opposite counterparts can work on the gradient at the same time and converge to a trade-off position which can balance the reconstruction performance and the informative of the decomposed signals. Then, since the gesture-specific classifier is the most crucial component in this approach, we train the $\mathcal{L}_g$ one more time in order to raise its priority. To sum up, in each training epoch, the $\mathcal{L}$ and $\mathcal{L}_g$ are optimized one time in turn. We adopt the Adam optimizer with learning rate e^{-4} for both loss functions. The algorithm is trained for 300 epochs. A dropout layer with 0.8 keep rate is added to the flatten layer in both classifiers in order to prevent overfitting.

6.5 The Novelty and Advantages

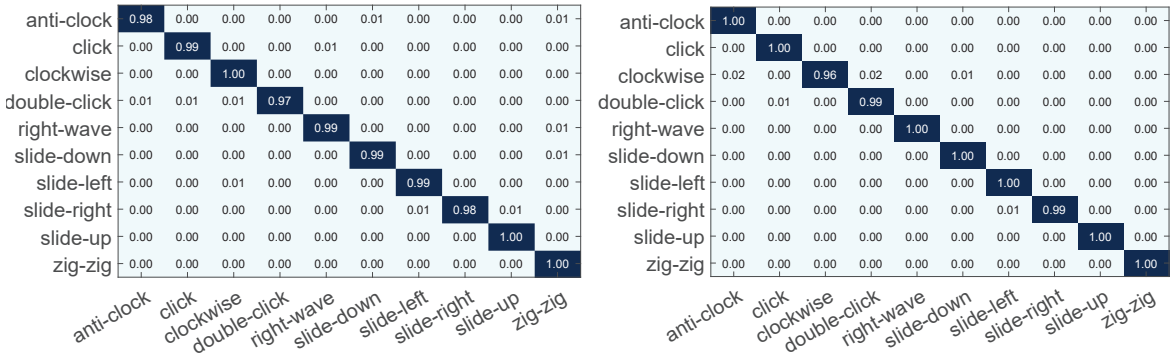
To the best of knowledge, we are the first to purify signal (such as gesture information) by combining signal decomposition and adversarial training. Our adversarial training strategy allows the classifier to focus on gesture-specific information and alleviate the influence caused by personal status (such as age and sex). Our classifier can be easily extended to mitigate other noise such as the influence caused by diverse phone types/brands (different hardware like transmitter and receiver).

In addition, the structure used in our classifier are fundamental neural network layers (e.g., convolutional, pooling, fully-connected layers): this is our advantage that achieves competitive performance using basic and elegant architecture. If necessary, it is easy to increase the depth of our framework in follow-up studies (i.e., straightforwardly replace the convolutional layers by more complex layer such as Convolutional Block Attention Module [33]) to obtain better performance at the cost of more expensive computation.

7 EVALUATION

7.1 Experimental Setup

We build StruGesture on a Samsung S5 mobile phone with Android 7.0 OS which is equipped with two microphones and two speakers. Our prototype uses one built-in speaker and two built-in microphones to transmit and receive the ultrasonic signals at a sample rate of 48 kHz. The received sound signal is sent to PC thorough Wi-Fi for further processing including demodulation, structure-borne separation and classification. The device used in our experiments is shown in Figure 12. All samples are collected under the **normal scenario**, in which user uses the index finger to slide at the central area of the phone's back with the normal force strength in the



(a) PD CM (b) PSD CM

Fig. 16. Confusion matrix of two situations

silent environment. The proposed deep learning model is trained on dual NVIDIA GeForce GTX 1080 Ti GPUs with 11 GB of GDDR5 memory.

Next, we report the hyper-parameter settings in detail. In the signal decomposition, the input signals have the shape $[M = 126, N = 38]$, the convolutional layer has 4 filers with size $[5, 5]$ and $[2, 2]$ strides. The following max-pooling has $[2, 2]$ window with $[2, 2]$ strides. In each de/convolutional layer, the activation function is ReLU and the padding method is 'SAME'. The deconvolutional layer has 1 filter and the other setting is the same.

The gesture- and user-specific classifiers have identical hyper-parameter setting: three convolutional layers have 16, 64, 256 filters with $[5, 5]$, $[3, 3]$, $[2, 2]$ sizes ($[1, 1]$ strides), respectively. All the kernel sizes and strides of the three pooling layers are $[2, 2]$. The two FC layers have 600 and 60 hidden neurons, respectively. The output layers of gesture- and user-specific classifiers have 6 and 9 neurons, respectively. The detailed data shape can be found in Figure 11.

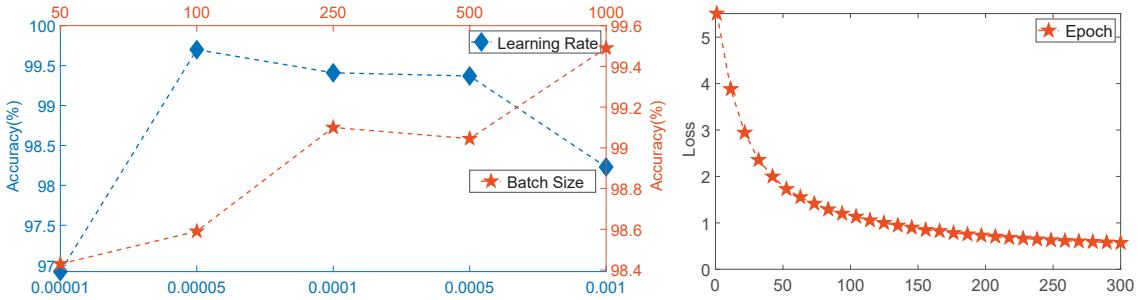
For evaluation, we recruit 10 volunteers (3 females and 7 males) aged from 23 to 30 years old. All volunteers are healthy graduate students and have no experience of gesture recognition. Before the experiment, we give a brief 10-minute introduction to each volunteer, explaining the regulations in data collection. All volunteers are asked to perform 10 gestures as shown in Figure 13 and repeat 100 times with index finger.

7.2 Recognition Accuracy

In this section, we focus on evaluating our approach over two situations, namely Person Semi-dependent (PSD) and Person Dependent (PD), compared with traditional classification methods (*i.e.*, Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbors (KNN), Convolution Neural Network (CNN)). In addition, the key parameters of the baselines are listed here: Linear SVM ($C = 1$), RF ($n = 50$), KNN ($k = 3$). The configuration of CNN baseline is the same as the gesture-specific component (introduced in Section 7.1). Person Semi-dependent situation denotes that some of the volunteer's data is seen in the training process and other unseen data will be tested. Person-dependent situation denotes that we just train part of one volunteer's dataset and then test his/her other unseen data. For Person Semi-dependent and Person Dependent, the dataset is randomly split into the training set (80%) and testing set (20%). Moreover, for PD, we only optimize the loss function L_g which represents the loss function of gesture-specific classifier, since there is only one subject which results in the failure of the signal decomposition and user-specific classifier. All data here is collected under normal office environment where there are people typewriting or talking around.

Figure 14 and 15 illustrate the comparison between our approach and other classifiers over 10 gestures and 2 situations. The results show that:

- The average recognition accuracy for PD, PSD situations are 98.9%, 99.5%, respectively.



(a) Accuracy vs. batch size & learning rate (b) Loss vs. Epoch

Fig. 17. Recognition accuracy under different parameters

- Our approach obviously outperforms all the baselines, including typical non-deep learning classifiers and competitive deep learning models, over all gestures and all situations.

Particularly, we find that the accuracy of our approach has a further improvement from PD to PSD while other methods have different levels of degradation. This is mainly because the increasing samples for PSD situation incur the intra-user noises which are various with different habits of users. While traditional methods are susceptible to these noises, our approach focus on combating them to improve the recognition accuracy. To have a closer observation, the Confusion matrix (CM) of the dataset relative to the person who has the average performance over 2 situations are reported in Figure 16.

To study the impact of parameters in terms of the classifier model, we examine the system performance by tuning three important parameters, *i.e.*, batch size, learning rate and epoch. Figure 17(a) shows the average recognition accuracy with different batch size and learning rate. The overall average accuracy over all tested samples are 98.4%, 98.6%, 99.1%, 99%, 99.5% for batch size of 50, 100, 250, 500, 1000, respectively. Similarly, for the learning rate varying from 0.00001 to 0.001, StruGesture achieves average accuracies of 96.9%, 99.7%, 99.4%, 99.5%, 98.2%, respectively. Figure 17(b) shows that the training loss degrades gradually with the increasing epochs and the degradation tends to converge after 200 epochs.

7.3 New Users Scenario

New users denote that the tested users does not appear in the training process. In this scenario, a total of 10 users participated in the evaluation. Each user performs 10 gestures, as shown in Figure 13, and 100 samples were collected for each gesture. Each user alternately acts as the new user, and all samples of the other 9 users are treated as a training set. Therefore, each of the 10 users will be tested once. Finally, the recognition accuracy for each gesture is the average result of all tested users..

7.3.1 Overall Accuracy: As shown in Figure 18, StruGesture achieves the average recognition accuracy of 73.1% for 10 gestures (*i.e.*, 0-shot). To further evaluate the performance under few-shot scenarios, in which only few new user's samples are trained together with previous training data, we put 10, 20, 30, 40, 50 samples into training set. The results show that StruGesture achieves the average recognition accuracies of 87.5%, 92.6%, 94.9%, 95.6% and 96.8%, respectively. This implies that 20 additional trained samples for a new user is sufficient to achieve the similar performance with the PSD situation.

7.3.2 User Numbers: We conduct extensive experiments to investigate the influence of subject amount on the system's performance. We evaluate our model over different number of users: the average recognition accuracy for 2, 4, 6, 8, 10 users are 78%, 91.4%, 91.6%, 92%, 92.6%, respectively. We also present the Receiver operating characteristic (ROC) curves in Figure 19. The larger Area Under Curve (AUC) indicates the higher recognition accuracy. We observe that StruGesture achieves better performance with increasing number of users and the

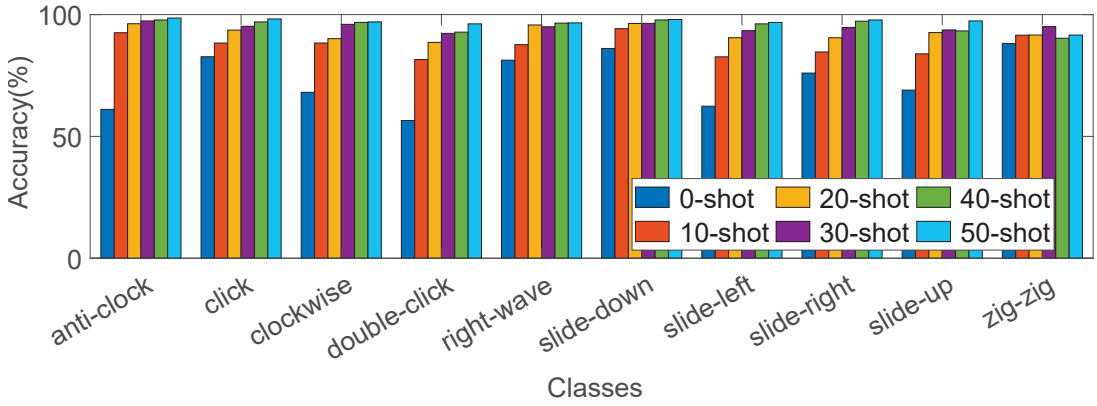


Fig. 18. Few-shot recognition results

performance tends to converge as the number of users is greater than 4. In this situation, we believe ten users are sufficient to evaluate the system's performance.

7.3.3 Layer Numbers: We further evaluate the effect of the number of decomposition layers. For simplification, we regard a de/convolutional layer and the following pooling layer as one layer. For example, as shown in Figure 11, the signal decomposition contains two layers (omit the combination layer as shown in Eq. 12): a convolutional layer (Eq. 8-9) and a deconvolutional layer (Eq. 10-11). We conduct experiments to evaluate the system performance while the signal decomposition has 2 (1 convolutional and 1 deconvolutional), 4 (2 convolutional and 2 deconvolutional), and 6 (3 convolutional and 3 deconvolutional) layers. All the hyper-parameters keep the same with Section 7.1.

We observe that the larger number of layers will lead to performance degradation. In detail, the average recognition accuracy for 2, 4, 6 layers are 92.6%, 88.5%, 88.8%, respectively. The ROC curve with the different number of decomposition layers are shown in Figure 20. One possible reason is that more decomposition layers will cause signal distribution distortions, and these distortions are difficult to be reconstructed in the following deconvolutional layers. Thus, we decide to use 2 layers in signal decomposition component.

7.4 Usage Scenarios

To better evaluate the performance of our system, we recruit a new volunteer for each scenario to collect 100 samples for each gesture. If not specified, for each new volunteer, we put his/her 20 samples collected under the normal scenario into the foregoing training set (*i.e.*, the training set in Section 7.3). The remaining 80 samples collected under respective scenario serve as the tested set (*i.e.*, 20-shot scenario).

7.4.1 Force Strength: StruGesture is robust over various force strength over different sliding times. We examine the impact of force strength on the gesture recognition with varied force strength applied on the back of mobile phone. We ask the volunteer to slide on the back surface with high and low strength, respectively. Figure 21 shows that StruGesture achieves the average recognition accuracies of 94.6% and 95.1% for the high and low strength, respectively. This indicates that the normalization process can effectively eliminate the negative influence of the force strength variance within different sliding times.

7.4.2 Finger Types: StruGesture is effective when performing gestures using different types of fingers. Although most gestures are performed with the index finger, users may use other infrequently used fingers to slide sometimes. To investigate the impact of different fingers, we ask the volunteer to perform gestures with the middle and thumb finger, respectively. Figure 22 shows that StruGesture achieves average recognition accuracies of 91.1%

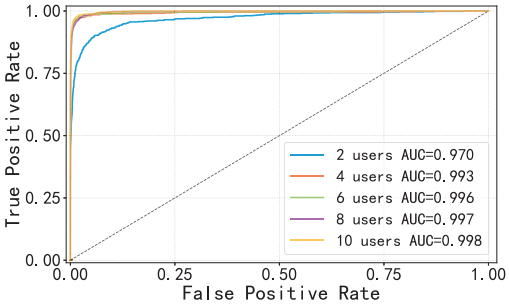


Fig. 19. ROC curve with different number of users

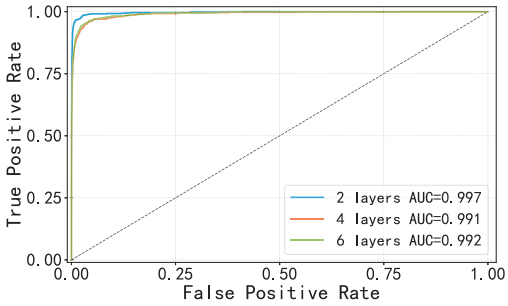


Fig. 20. ROC curve with different number of layers

and 95.4% for these two scenarios, which demonstrate the robustness of our system. This is mainly because the sliding habits, including the sliding trace and speed, are stable for the same person despite different fingers.

7.4.3 Sliding Area: StruGesture keeps high recognition accuracy with different sliding areas. Besides the central area of the phone, user usually slides in various areas. This implies that our system should tolerate the sliding area diversity. We ask the volunteer to perform gestures in the upper and lower area, respectively. As shown in Figure 23, our system achieves the average recognition accuracies of 96.3% and 97.1%, respectively. This result verifies the structure-borne propagation model illustrated in section 3, which points out that the sliding pattern is independent of sliding areas.

7.4.4 Close-in Disturbance: StruGesture is robust against the interference from the close-in hand movement. In this scenario, while the user's index finger slides on the back surface, the middle finger moves in the air meanwhile. In this way, the close-in disturbance mainly comes from the palm and other fingers when performing the gestures. To make a fair comparison between our system and the air-borne propagation based scheme, we retrained the air-borne data extracted from the same hybrid CIR signals. Figure 24 shows that our system (90.9%) achieves high improvement than the air-borne scheme (66.4%). This is owing to the fact that the structure-borne propagation of ultrasound signals is immune to the obstruction in the air.

7.4.5 Sweaty&Gloved: StruGesture achieves the average recognition accuracies of 97% and 87.3% for the sweaty and gloved finger, respectively. It's annoying to the user when the finger is sweaty or wearing glove since the touchscreen can't respond well. To demonstrate the performance of our system, we instrument the volunteer to perform all gestures with sweaty and gloved finger, respectively. Figure 25 shows that our system is robust to the sweaty finger while the performance with glove degrades by nearly 10%. The accuracy drop is mainly due to lack of the diversity from the naked finger to glove, which can be improved by involving few gloved samples in training process.

7.4.6 Sound Interference: StruGesture is robust to background audible sound interference. It is a familiar usage scenario that user is using StruGesture when there is background sound around (*i.e.*, Out-sound) or there is self playing sound such as music (*i.e.*, In-sound). We use shortened form Out-sound and In-sound to denote these two scenarios, respectively. To explore the performance under the scenarios above, we conduct two experiments respectively. For Out-sound scenario, we put another phone 1 *m* away to play music (around 70 dB). For In-sound scenario, we add the sound signals to the passband signals before transmitting. Figure 26 shows that our system achieves the average recognition accuracies of 92.6% and 84.4% for Out-sound and In-sound scenarios, respectively. It indicates that the background sound has no influence on our system. This is mainly because the frequency of audible sound is much lower than the ultrasound. In contrast, the accuracy of In-sound scenario has a drop of more than 8% . This is due to the minor vibration incurred by self sound playing.

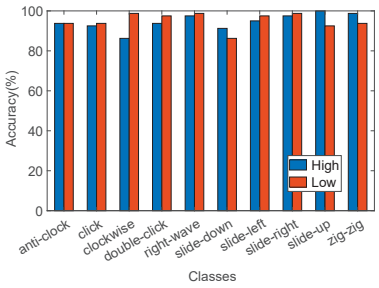


Fig. 21. Accuracy with different force strength

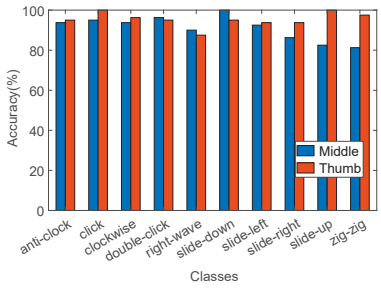


Fig. 22. Accuracy with different types of finger

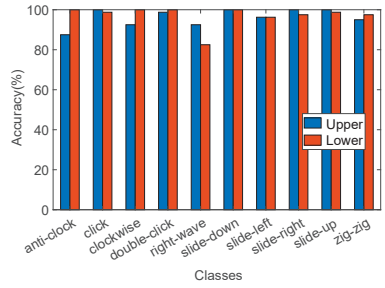


Fig. 23. Accuracy with different areas

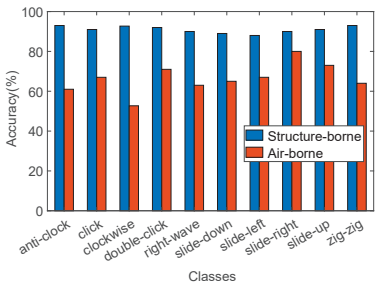


Fig. 24. Accuracy with disturbance

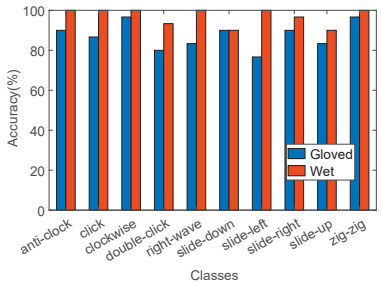


Fig. 25. Accuracy with gloved&wet finger

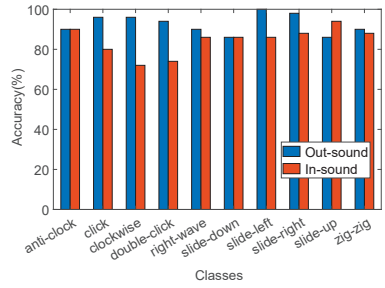


Fig. 26. Accuracy with playing music

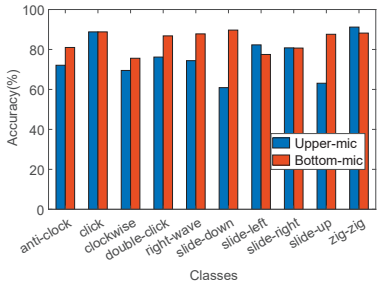


Fig. 27. Accuracy with different microphones

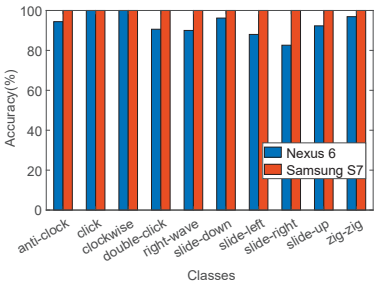


Fig. 28. Accuracy with different types of phones

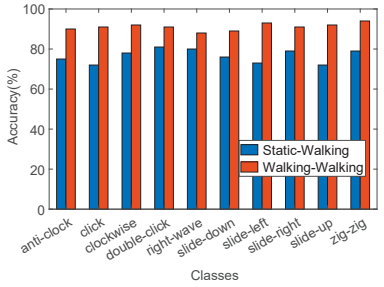


Fig. 29. Accuracy with user walking

7.4.7 Microphone Numbers: Combining the data from two microphones is necessary in our system. To evaluate the impact of microphone numbers, we extract the data corresponding to upper and lower microphones from the mixed data used in Section 7.2. Then, we retrained and retested these two data using the same method with 20-shot scenario. Figure 27 shows that the average recognition accuracies relative to the upper and lower microphones are 84.4% and 75.9%, respectively. In comparison, the average accuracy is 92.6% with the dual input, as shown in Figure 18. The performance drops since different gestures may have the similar pattern for the same microphone. For example, sliding left and sliding right gestures are difficult to distinguish for the upper microphone.

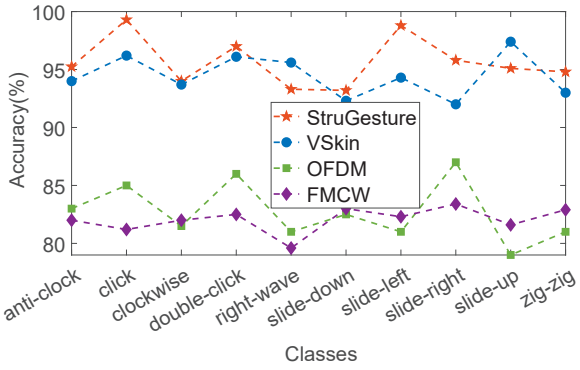


Fig. 30. Accuracy for different de/modulation schemes

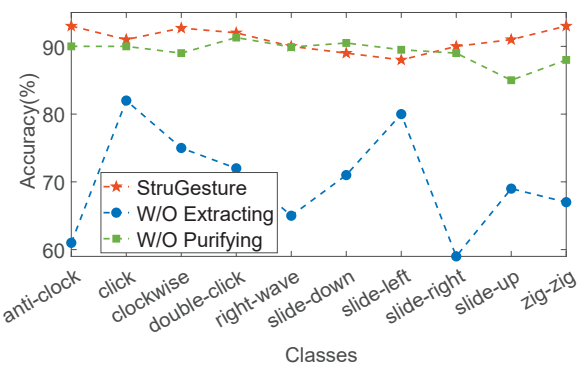


Fig. 31. Accuracy for the ablation study

7.4.8 Phone Types: StruGesture works well for different types of mobile phones. We additionally perform gesture recognition on Samsung Galaxy S7 and Nexus 6, respectively. Note that the collected data should be retrained since the layout of microphones and speakers are diverse over different types of phones. Additionally, the evaluation under this scenario is executed in the PD scenario. As shown in Figure 28, StruGesture achieves the average recognition accuracies of 100% and 93.1% for S7 and Nexus 6, respectively. Despite that we find the noise of Nexus 6 is considerably larger than S7, the recognition performance for Nexus 6 is good enough to meet the application requirement.

7.4.9 User Walking: StruGesture has a slight degradation of the recognition accuracy when the user is walking. To evaluate the walking performance, we train the PD model with the data when the user is in walking and static states under the normal scenario. All testing data comes from the walking state. As shown in Figure 29, StruGesture achieves an average recognition accuracy of 91.1% for the walking state, which is lower than 98.9% illustrated in Section 7.2. It is mainly due to the violent shaking of the hand when the user is on the move. Additionally, we combine the data under the static and walking states to study the data compatibility. The static state data serves as the training data, and the data in the walking state serves as the testing data. Figure 29 shows that StruGesture achieves an average recognition accuracy of 76.5%, which is not enough for everyday use.

7.4.10 De/modulation Schemes: Our results show that our de/modulation schemes are superior to other typical schemes. To make a fair comparison, all signals only differ in modulation and demodulation schemes. We use the PD model to train and test the collecting data under the normal scenario. As shown in Figure 30, the average recognition accuracies for StruGesture, VSkin, OFDM, FMCW are 95.7%, 94.5%, 82.7%, 82.1%, respectively. Our de/modulation scheme's performance is much better than that of traditional modulation schemes (*i.e.*, OFDM, FMCW). Although our scheme's performance is similar to VSkin, the time consumption of our scheme is only one fifth of that of VSkin, as illustrated in Table 1.

7.4.11 Ablation Study: Structure-borne component extracting and purifying are both essential to our system. We use the same collecting data under the "Close-in Disturbance" scenario to evaluate the system performance without signal extracting (abbreviated as W/O Extracting) and purifying (abbreviated as W/O Purifying), respectively. Figure 31 shows that the system's recognition accuracies are 70.1%, 89.2% for W/O Extracting, W/O Purifying, respectively. It means that the system's recognition accuracy without signal extracting degrades by 20.1% compared with StruGesture. This is because the critical advantage of signal extracting is to reduce air-borne interference. Additionally, the system without signal purifying has a slight degradation compared with the original system. This is mainly because the static component of the structure-bore signals may be slightly different when the user holds the phone at different times. The purifying operation can thus improve the system performance by removing the effect of the static signals.

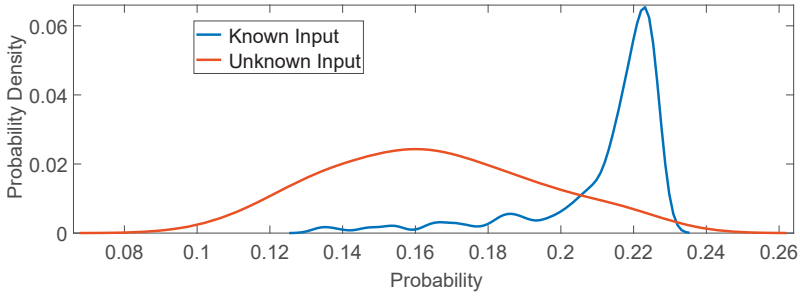


Fig. 32. Probability distribution for the known input and unknown input

7.4.12 Unknown Input: In the real-world deployment, it is important to have the ability to detect the unknown input (i.e., the gestures not included in training set). To address this issue, we examine the predicted probability (after the softmax activation function) in the last layer of gesture-specific classifier. The output layer contains 10 positive scores (as we have 10 gestures) representing the probability that the input sample fall into each of the 10 gestures. If the highest probability is lower than a predefined threshold, our model regards the input sample is not similar to any of the 10 gestures and recognize it as unknown gesture. Our hypothesis is that if the input gesture never appears in the training phase, its pattern will not match with any training gestures (i.e., the 10 gestures appeared in training stage) and its probability is small in all categories.

We conduct extensive experiments to evaluate the ability of our model detecting the unknown input. We train our model by 10 gestures, then we feed 200 unseen samples (from 2 gestures, 100 samples writing number '3' while 100 samples writing number '9') into well-trained model for recognition. We provide density of the highest predicted probability of known gestures (the 10 gestures appeared in training) and unknown gestures (the 2 unseen gestures) and observe a different distribution (Figure 32). This demonstrates our previous hypothesis is meaningful.

By setting the threshold as 0.18, StruGesture can correctly recognize 141 out of 200 unknown input, claiming an accuracy of 70.5%, demonstrating that our system is highly robust and able to screen the unknown gesture. As a price, the recognition accuracy of known gestures has a slight drop from 92.6% to 91.9%.

8 LIMITATIONS AND DISCUSSION

StruGesture proves the feasibility of using structure-borne propagation to recognize sliding gestures with high accuracy. However, StruGesture has some limitations in the current form of implementation. Firstly, the gesture waving speed should be in the normal range, which can not be too fast or slow. Too fast motion will incur the doppler effect, which impacts the baseband signals and the subsequent structure-borne measurement. Besides, the gesture pattern with too slow motion is deviated from the overall pattern severely, which may disable our classifier. We leave the performance study with wide range of waving speed as future work. Secondly, although different mobile phones have the similar layout, the distance between the speaker and microphones may be slightly different. To enable StruGesture, we need to retrain the collected data for each type of phone, which reduces universality and incurs the computation overhead. Thirdly, the classification process is realized offline currently. In the future, we will consider embedding the training model generated by the server into the mobile phones to enable the realtime recognition. Furthermore, although our model (i.e., the gesture-specfic classifier) achieves recognition performance of above 90% in most scenarios, the user-specific classifier has a lower user recognition accuracy (62.4%). This is acceptable as user identification is not a main task of this work. In fact, the difference among distinct gestures are obviously greater than the difference among various users, because the former is caused by gesture trajectories and the latter are mainly caused by the velocity. One future research scope of our research is to increase user recognition's ability, thus enabling StruGesture's capacity in privacy and security.

9 CONCLUSION

In this paper, we make the following three key contributions. Firstly, we propose the structure-borne propagation model on the back of mobile phone, which builds the correlation between the structure-borne path change and the sliding gestures. Secondly, we propose a systematic approach to extract the structure-borne component. Thirdly, we develop a new deep learning algorithm to learn the gesture-specific representation for recognition. Our results show that StruGesture achieves an average classification accuracy of 99.5%.

ACKNOWLEDGMENTS

This research is supported by NSFC A3 Project 62061146001, PKU-Baidu Funded Project 2019BD005, PKU-NTU collaboration Project. In part by National Natural Science Foundation of China(Grant No. 12071460). In part by National Natural Science Foundation of China (Grant No. 61802007, 62022005).

REFERENCES

[1] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. 2015. WiGest: A Ubiquitous WiFi-based Gesture Recognition System. In *Proceedings of IEEE INFOCOM*.
[2] Kamran Ali, Alex X. Liu, Wei Wang, and Muhammad Shahzad. 2015. Keystroke Recognition Using WiFi Signals. In *Proceedings of ACM MobiCom*.
[3] Md Tanvir Islam Aumi, Sidhant Gupta, Mayank Goel, Eric Larson, and Shwetak Patel. 2013. DopLink: using the doppler effect for multi-device interaction. In *Proceedings of ACM UbiComp*.
[4] Ke-Yu Chen, Daniel Ashbrook, Mayank Goel, Sung-Hyuck Lee, and Shwetak Patel. 2014. AirLink: sharing files between multiple devices using in-air gestures. In *Proceedings of ACM UbiComp*.
[5] Mingshi Chen, Panlong Yang, Jie Xiong, Maotian Zhang, Youngki Lee, Chaocan Xiang, and Chang Tian. 2019. Your Table Can Be an Input Panel: Acoustic-based Device-Free Interaction Recognition. In *Proceedings of ACM UbiComp*.
[6] Emilio Granell and Luis A Leiva. 2017. βTap: back-of-device tap input with built-in sensors. In *Proceedings of ACM MobileHCI*.
[7] Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. Soundwave: Using the Doppler Effect to Sense Gestures. In *Proceedings of ACM CHI*.
[8] Chris Harrison, Julia Schwarz, and Scott E Hudson. 2011. TapSense: enhancing finger interaction on touch surfaces. In *Proceedings of ACM UIST*.
[9] Franz Hlawatsch and François Auger. 2008. *Time-frequency analysis*. Wiley Online Library.
[10] Eva Kollorz, Jochen Penne, Joachim Hornegger, and Alexander Barke. 2008. Gesture recognition with a time-of-flight camera. *International Journal of Intelligent Systems Technologies and Applications* 5, 3 (2008), 334.
[11] Huy Viet Le, Sven Mayer, Patrick Bader, and Niels Henze. 2017. A smartphone prototype for touch interaction on the whole device surface. In *Proceedings of ACM MobileHCI*.
[12] Kang Ling, Haipeng Dai, Yuntang Liu, Alex X Liu, Wei Wang, and Qing Gu. 2020. Ultragesture: Fine-grained gesture sensing and recognition. *IEEE Transactions on Mobile Computing* (2020).
[13] Jian Liu, Yingying Chen, Marco Gruteser, and Yan Wang. 2017. VibSense: Sensing Touches on Ubiquitous Surfaces through Vibration. In *Proceedings of IEEE SECON*.
[14] Jian Liu, Chen Wang, Yingying Chen, and Nitesh Saxena. 2017. VibWrite: Towards finger-input authentication on ubiquitous surfaces via physical vibration. In *Proceedings of ACM CCS*.
[15] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. In *Proceedings of ACM CHI*.
[16] 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 ACM SenSys*.
[17] Corey R Pittman and Joseph J LaViola Jr. 2017. Multiwave: Complex Hand Gesture Recognition Using the Doppler Effect. In *Proceedings of ACM GI*.
[18] A Rodríguez Valiente, A Trinidad, JR García Berrocal, C Górriz, and R Ramírez Camacho. 2014. Extended high-frequency (9–20 kHz) audiometry reference thresholds in 645 healthy subjects. *International journal of audiology* 53, 8 (2014), 531–545.
[19] 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 ACM UbiComp*.
[20] Huihuang Zhang Lili Qiu Sangki Yun, Yi-chao Chen and Wenguang Mao. 2017. Strata: Fined-Grained Device-Free Tracking Using Acoustic Signals. In *Proceedings of ACM MobiSys*.

[21] AlsoRF Schaufele and T Shimanouchi. 1967. Longitudinal acoustical vibrations of finite polymethylene chains. *The Journal of Chemical Physics* 47, 9 (1967), 3605–3610.
[22] Shaikh Shawon Arefin Shimon, Sarah Morrison-Smith, Noah John, Ghazal Fahimi, and Jaime Ruiz. 2015. Exploring user-defined back-of-device gestures for mobile devices. In *Proceedings of ACM MobileHCI*.
[23] Ke Sun, Wei Wang, Alex X Liu, and Haipeng Dai. 2018. Depth aware finger tapping on virtual displays. In *Proceedings of ACM MobiSys*.
[24] 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 ACM MobiCom*.
[25] 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 ACM MobiCom*.
[26] Sheng Tan and Jie Yang. 2016. WiFinger: Leveraging Commodity WiFi for Fine-grained Finger Gesture Recognition. In *Proceedings of ACM Mobihoc*.
[27] Yu-Chih Tung and Kang G Shin. 2016. Expansion of human-phone interface by sensing structure-borne sound propagation. In *Proceedings of ACM MobiSys*.
[28] Chong Wang, Zhong Liu, and Shing-Chow Chan. 2014. Superpixel-based hand gesture recognition with kinect depth camera. *IEEE transactions on multimedia* 17, 1 (2014), 29–39.
[29] Lei Wang, Ke Sun, Haipeng Dai, Alex X Liu, and Xiaoyu Wang. 2018. WiTrace: Centimeter-Level Passive Gesture Tracking Using WiFi Signals. In *Proceedings of IEEE SECON*.
[30] Wei Wang, Alex X. Liu, and Ke Sun. 2016. Device-free gesture tracking using acoustic signals. In *Proceedings of ACM MobiCom*.
[31] Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. 2007. Lucid touch: a see-through mobile device. In *Proceedings of ACM UIST*.
[32] Pui Chung Wong, Hongbo Fu, and Kening Zhu. 2016. Back-Mirror: back-of-device one-handed interaction on smartphones. In *Proceedings of ACM SA*.
[33] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. Cbam: Convolutional block attention module. In *Proceedings of ECCV*. 3–19.
[34] Xiang Xiao, Teng Han, and Jingtao Wang. 2013. LensGesture: augmenting mobile interactions with back-of-device finger gestures. In *Proceedings of ACM ICMI*.
[35] Nan Yu, Wei Wang, Alex X Liu, and Lingtao Kong. 2018. QGesture: Quantifying gesture distance and direction with WiFi signals. In *Proceedings of ACM UbiComp*.
[36] Sangki Yun, Yi-Chao Chen, and Lili Qiu. 2015. Turning a mobile device into a mouse in the air. In *Proceedings of ACM MobiSys*.
[37] Cheng Zhang, Aman Parnami, Caleb Southern, Edison Thomaz, Gabriel Reyes, Rosa Arriaga, and Gregory D Abowd. 2013. BackTap: robust four-point tapping on the back of an off-the-shelf smartphone. In *Proceedings of ACM UIST*.
[38] Maotian Zhang, Panlong Yang, Chang Tian, Lei Shi, Shaojie Tang, and Fu Xiao. 2015. Soundwrite: Text input on surfaces through mobile acoustic sensing. In *Proceedings of ACM SmartObjects*.