Push the Limit of Acoustic Gesture Recognition

Yanwen Wang, *Member, IEEE*, Jiaxing Shen, and Yuanqing Zheng, *Member, IEEE*

Abstract—With the flourish of the smart devices and their applications, controlling devices using gestures has attracted increasing attention for ubiquitous sensing and interaction. Recent works use acoustic signals to track hand movement and recognize gestures. However, they suffer from low robustness due to frequency selective fading, interference and insufficient training data. In this work, we propose RobuCIR, a robust contact-free gesture recognition system that can work under different practical impact factors with high accuracy and robustness. RobuCIR adopts frequency-hopping mechanism to mitigate frequency selective fading and avoid signal interference. To further increase system robustness, we investigate a series of data augmentation techniques based on a small volume of collected data to emulate different practical impact factors. The augmented data is used to effectively train neural network models and cope with various influential factors (e.g., gesture speed, distance to transceiver, *etc.*). Our experiment results show that RobuCIR can recognize 15 gestures and outperform state-of-the-art works in terms of accuracy and robustness.

Index Terms—Acoustic sensing, smart devices, gesture recognition, contact-free, data augmentation

1 Introduction

Motivation. Contact-free gesture recognition techniques facilitate human-computer interaction (HCI) methods. They enable users to control digital devices without any physical contact. Imagine that we may simply perform a gesture nearby a smart speaker at home to switch music or control speaker volume while chatting in the car. We could block an incoming call in meeting without touching the device, or enable contact-free human computer interaction in virtual and augmented reality applications. These contact-free systems provide immersive user experience and support a variety of novel applications in gaming, smart home, and healthcare. For example, contact-free gesture recognition provides more immersive user experience when playing VR/AR games. Contact-free gesture recognition can be useful for smart devices, especially when operating with touchscreens appears to be particularly inconvenient (e.g., wearing gloves, devices in pocket). Contact-free user interaction can also be applied in kiosks in public area to reduce the risk of spreading germs via touch screens. Such applications require high accuracy and robustness in various application scenarios. In this paper, we aim to design a contact-free gesture recognition system that can achieve accurate and robust gesture recognition.

Prior Works and Limitation. Existing RF-based HCI technologies explore the potential of controlling devices using wireless signals [2], [14], [27], [44]. Such technologies require specialized hardware (e.g., Universal Software Radio Peripheral (USRP) [14], [27], Frequency Modulated Continuous Wave (FMCW) radar [2]), which incurs high costs and prohibits a wide deployment.

Recent acoustic sensing systems leverage speakers and microphones, embedded in smart devices, to enable contact-free motion tracking [17], [18], [22], [43], [49]. FingerIO [22] is able to accurately track moving objects (e.g., a waving hand) by transmitting Orthogonal Frequency Division Multiplexing (OFDM) modulated acoustic signals and analyzing the signal variations caused by the moving object. LLAP [43] is able to track finger movements by measuring the phase change of the received signals. Strata [49] achieves a higher accuracy in tracking one moving object by estimating the Channel Impulse Response (CIR) of the reflected signal.

Those works model the whole finger/hand as *a single reflection point* and intentionally neglect weak multi-path signals. Note that such a single reflection model can effectively enhance its performance in tracking one moving object. Yet, modeling a hand as a single reflection point cannot provide sufficient resolution for gesture recognition due to relatively complex finger movements. For instance, in order to recognize spread or pinch gesture (illustrated in Fig. 1), we need to differentiate and track five fingers simultaneously.

Since it is very hard to accurately model the complex signal reflections, recent works attempt to leverage neural networks to automatically extract effective features from received signals [13], [17]. For example, UltraGesture [17] uses a deep neural network to extract features from measured CIR magnitude for identifying different gestures. However, due to insufficient training data, the trained model cannot handle various real practical impact factors in practice.

Challenges. Implementing a robust acoustic gesture recognition system is a non-trivial task due to complicated movements of fingers. One challenging issue of acoustic based gesture recognition is frequency selective fading (FSF) due to the multi-path transmissions of acoustic signals as well as the speaker and microphone distortion at high frequencies (e.g., $\geq$ 18KHz). Previous work only sends an acoustic signal at a fixed frequency [17], which may experience

- *Yanwen Wang is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, and also with Hunan University, Changsha 410082, China. E-mail: yanwen.wang@connect.polyu.hk.*
- *Jiaxing Shen and Yuanqing Zheng are with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. E-mail: jiaxing.shen@connect.polyu.hk, csyqzheng@comp.polyu.edu.hk.*

Manuscript received 12 Mar. 2020; revised 13 Oct. 2020; accepted 15 Oct. 2020. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Yuanqing Zheng.)
Digital Object Identifier no. 10.1109/TMC.2020.3032278

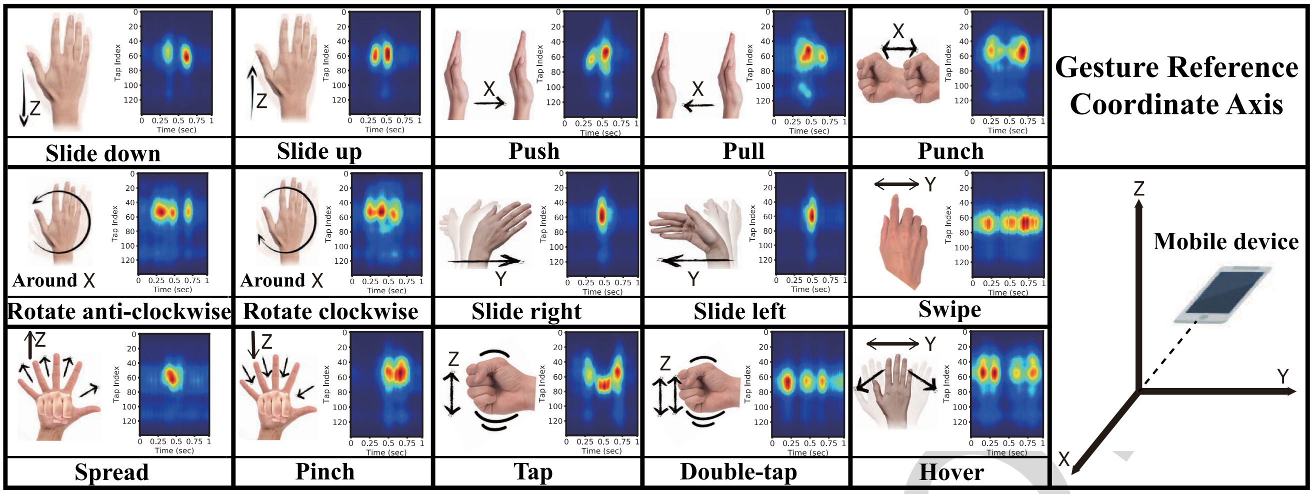


Fig. 1. 15 types of hand gestures and their corresponding CIR patterns. To standardize the tested gestures, we divide the test gestures into different categories including (1) the typical gestures involving hand movements along 3 axes in 3D space (slide down/up (Z-axis), push/pull (X-axis), and slide right/left (Y-axis)); (2) rotation around an axis; and (3) some complex hand gestures (punch, spread, pinch, swipe, tap, double taps, and hover). To better depict the test gestures, in the figure, we use $\xrightarrow{X}$ to represent a hand movement along an axis (e.g., X axis), and use a double-headed arrow (e.g., $\overset{X}{\leftrightarrow}$) to represent a back-and-forth movement (e.g., punch) along the axis.

dramatic fading in signal magnitude in particular environments. Intuitively, one can simultaneously transmit acoustic signals at multiple frequencies to alleviate the impact of FSF and the signal distortion at high frequencies. However, the computational cost involved in processing the multi-frequency signal is high and prohibitive to meet real-time processing requirement on lightweight smart devices (e.g., smart watch).

Another practical challenge arises from insufficient training data. To ensure robust gesture recognition, the neural network requires sufficient training data to cover different variations of gestures under diverse practical scenarios [48]. In practice, it is inconvenient and sometimes impractical to collect sufficient training data from users.

Our Solution. We propose RobuCIR, a robust gesture recognition system based on acoustic signals transmitted by the smartphone, which achieves high recognition accuracy under various practical impact factors. RobuCIR can identify 15 standardized gestures, as illustrated in Fig. 1. RobuCIR can detect a gesture ranging up to approximately 50cm from the smartphone.

In our solution, we adopt frequency hopping to mitigate FSF and carefully design low pass filters to avoid inter-subframe interference (described in Section 3.2). In particular, we modulate a known baseband signal, up-convert to different frequencies, and transmit at each frequency periodically. We regard this periodical signal as a channel measurement frame, which consists of multiple subframes at different frequencies. To further enhance the robustness of RobuCIR, different from prior work that only exploits the magnitude component, we synthetically consider both magnitude and phase components to capture more information of the multi-path. We notice that the phase component is generally more robust to interference and noise, which is promising to achieve high accurate localization and tracking [4], [43], [49].

To address the challenge of lacking of training data, instead of manually collecting all training data, we collect a small amount of raw data and apply a series of selective data augmentation techniques to enhance the data. Such well-orchestrated data augmentation techniques come from our key observation that the variations of the CIR measurements under different practical impact factors (e.g., different gesture speeds, distance to transceiver, Non-Line-of-Sight (NLOS), noises) generate different patterns, which are traceable and correlate to the gesture variations. RobuCIR thus can handle various practical impact factors which may not be fully captured by the raw data but by the augmented data. To the best of our knowledge, we are the first to correlate the variations of CIR measurements with different practical impact factors.

Different gestures generate different CIR images with different patterns, as shown in Fig. 1, which are estimated by Least Square (LS) channel estimation technique. To identify gestures, motivated by recently impressive performance on image classification, we train a classifier using neural networks via supervised learning. In specific, our classifier consists of a Convolutional Neural Network (CNN) and a Long-Short Term Memory (LSTM) network to automatically extract complicated features from the augmented data and perform gesture recognition.

Evaluation. We implement all functional components including signal processing, data augmentation and coupled deep learning architecture and conduct extensive evaluation in various experiment settings. We transmit the signal at three different frequencies to eliminate the frequency selective fading and conduct ten-fold cross-validation with the data collected by various types of smartphones. In our experiment, RobuCIR achieves 98.4 percent recognition accuracy under various practical impact factors in the task of recognizing the 15 gestures.

Our Contributions. Such a holistic design allows us to achieve higher channel measurement resolution and sufficient training data, while meanwhile mitigating FSF and ISI without posing extra computational overhead on lightweight smart devices. In our experiment, RobuCIR achieves 98.4 percent recognition accuracy under various practical impact factors in the task of recognizing the 15 gestures.

We make the following contributions:

- We address the challenge of frequency selective fading caused by multipath effect by periodically transmitting the acoustic signals with different frequencies.
- We leverage the correlation of the CIR measurements and gesture variations to overcome the challenge of insufficient training data. The augmented data is automatically generated without user involvement.
- We implement RobuCIR and conduct extensive evaluation. The experiment results show that RobuCIR outperforms state-of-the-art work in terms of accuracy and robustness under various practical impact factors.

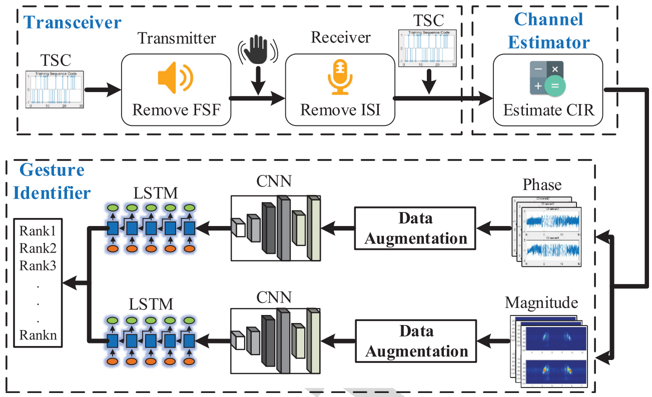


Fig. 2. Overview of RobuCIR.

2 Background

2.1 Channel Measurement

Channel measurements determine the fading and path loss of the wireless channel. Channel measurements are represented with complex values, in which two key parameters, signal strength and signal phase, can be measured. The signal strength indicates the signal fading while the signal phase reveals the propagation delay and distance. As human gestures could influence the wireless channel, channel measurements may involve the unique pattern of certain gestures, which can be used to infer the gesture types.

2.2 Channel Impulse Response

Existing acoustic signal based gesture recognition systems detect the finger/hand movement by measuring the CIR of the reflected signal frames. The transmitter modulates a known signal, up-converts to a high frequency f_c, and continuously sends this inaudible audio signal frame. The frame is then reflected from a moving finger/hand and received by the receiver. The received frame is down-converted to generate an imaginary and real components of the baseband signal.

The acoustic channel can be modeled as a Linear Time-Invariant system, which is effective to model propagation delay and signal attenuation along multiple propagation paths. The received signal can be mathematically represented as $r[n] = s[n] * h[n]$, where $h[n]$ represents CIR of the acoustic channel, $r[n]$ and $s[n]$ represent the received signal and transmitted signal, respectively.

In practice, one may estimate the CIR by sending a known signal frame as a probe. With the received frame, Least Square channel estimation method can estimate CIR [17], [49]. In particular, LS channel estimation measures the channel $h = \arg\min_{h} \|r - Mh\|^2$, where M is the training matrix consisting of transmitted circulant orthogonal codes (e.g., training sequence code (TSC) [49], Barker code [17]). CIR measurement is represented with a set of complex values, in which each complex value measures the channel information of a certain propagation delay range and the corresponding amplitude and phase of the CIR can be obtained.

2.3 Frequency Selective Fading

In wireless communication and acoustic sensing, the emitted signal experiences reflections from objects (e.g., ground, wall, desks, chairs) in the environment, which results in multipath signals with similar strength in the air. Such multipath signals might be destructively added together (e.g., two signals with phase variation of π) and cause cancellation of certain frequencies at the receiver, which results in deep nulls in the received signal strength. FSF could significantly affect the signal patterns caused by the gestures and, hence, degrade the performance of the gesture recognition systems if we cannot handle it properly.

3 System Design

3.1 Overview

Fig. 2 illustrates the overview of RobuCIR. RobuCIR consists of three main components, which are *Transceiver*, *Channel Estimator* and *Gesture Identifier*. In *Transceiver*, a speaker plays an inaudible frame for channel measurement and a microphone records the received frame. Within each inaudible frame, the carrier frequency hops among multiple frequencies to mitigate FSF. Then, *Channel Estimator* estimates the CIR with the LS channel estimation. Finally, *Gesture Identifier* regards CIR phases and magnitudes measured across a certain time as a *CIR phase image* and a *CIR magnitude image*, respectively. To improve the robustness of our system, we perform data augmentation on each *CIR image* so that the augmented data can cover various real practical impact factors. As such, the final model trained with augmented data can cope with various factors (e.g., gesture speed, distance, noise, *etc.*). In particular, the augmented data are used to train a CNN to automatically extract features and an LSTM network to perform gesture recognition.

3.2 Design of Transceiver

Fig. 3 illustrates the design of transceiver. The transceiver consists of a speaker acting as an acoustic transmitter and a microphone acting as a receiver, which are collocated and synchronized in a single device. The transmitter sends a pre-defined signal frame and the receiver measures the CIR by analyzing the received signal frame [17], [49]. In particular, the transmitter sends a 26-bit Training Sequence Code that has good autocorrelation property and facilitates channel measurements [37]. The TSC are then up-sampled and up-converted to the carrier frequency f_c before transmission. To ensure the transmitted frame are inaudible, the carrier frequency is set to be higher than 18 KHz (i.e., $f_c \geq 18$ KHz). To avoid inter-subframe interference (ISI), previous

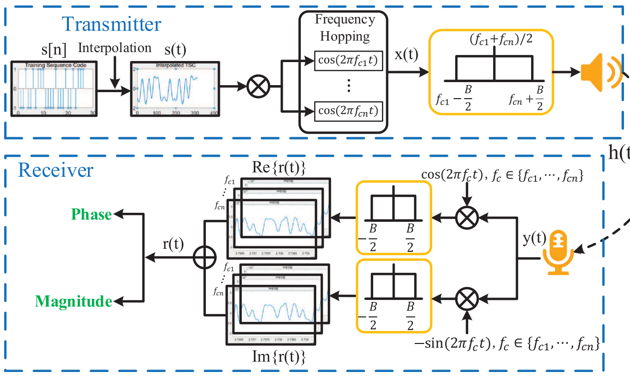


Fig. 3. Design of transceiver.

works add guard intervals (GI) between frames. In particular, zero samples are added between frames so that the echoes of current frame would not be mixed in the following frames.

3.2.1 Mitigate Frequency Selective Fading

Existing works modulate and up-convert the pre-defined TSC symbols to a single frequency. Single-frequency based method may suffer from FSF, since the audio signals reflected from multiple objects may add up destructively with each other, which greatly decreases the system performance.

To better understand how FSF influences the channel measurements, we conduct experiments and measure the CIR magnitude and phase when transmitting at multiple frequencies. In the experiment, we perform push and pull gestures 5 times in front of the transceiver. We send the BPSK modulated TSC at three frequencies.

Fig. 4 shows the CIR magnitudes measured during the experiment. In the figure, X-axis represents time, while Y-axis represents CIR tap positions. The brightness represents the CIR magnitude. Each tap corresponds to a certain delay range and reflected signals with similar propagation delays are summarized in the same tap. In Fig. 4, when transmitting at f_{c1} (upper panel), the CIR magnitude changes substantially due to pull and push activities. When transmitting at f_{c2} (mid panel), due to frequency selective fading, the CIR magnitude dramatically decreases and exhibits less clear patterns. Similar to the influence on CIR magnitude, frequency selective fading also affects the phase measurements at different frequencies. The experiment results indicate that the frequency selective fading, if not handled properly, could dramatically influence the channel measurement results, leading to low accuracy and degraded robustness in gesture recognition.

Along with the magnitude, we can also obtain the phase information from the CIR measurements. We conduct another experiment where we move a cardboard near the transceiver. First, the cardboard keeps static for around 5s and then moves backward for around 5s along a straight line at a constant speed. Note that the hardware of transmitter and receiver introduce constant phase offset throughout the experiment, which therefore can be removed by calculating the phase difference between two adjacent phase measurements (discussed in detail in Section 3.2.3). Figs. 5a and 5b plot the measured phase values when transmitting

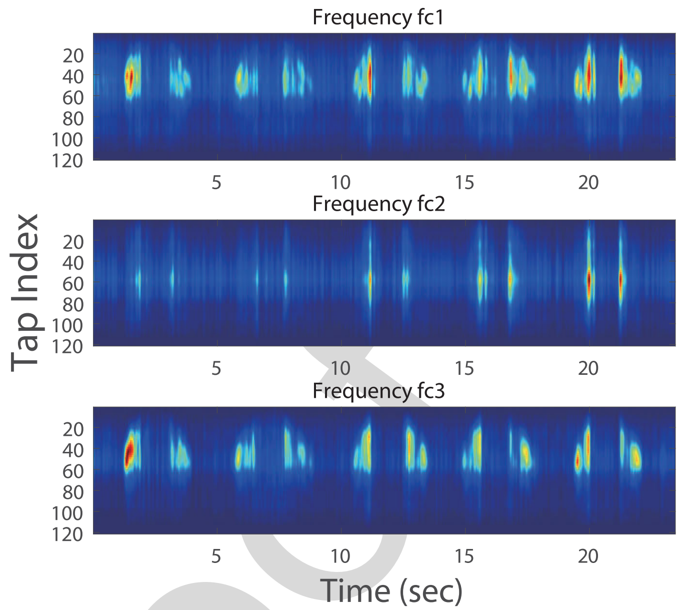
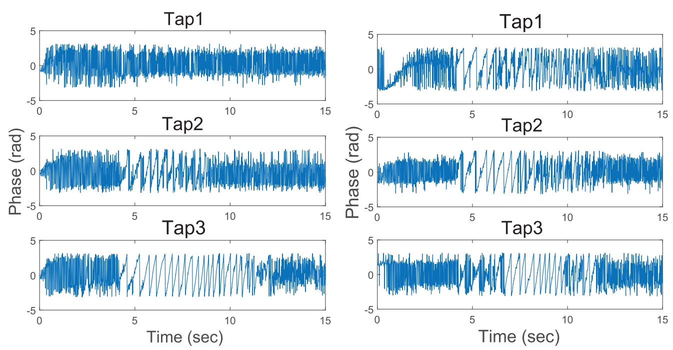


Fig. 4. CIR when performing push and pull.

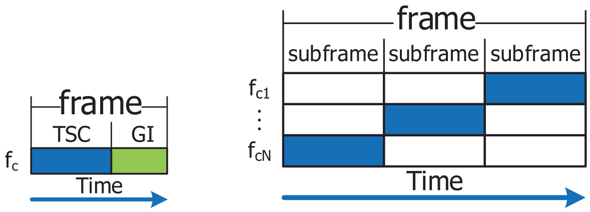
at f_{c1} and f_{c2}, respectively. Among all taps, only three taps are plotted for better illustration. We observe the linearly increasing pattern in some taps as path length increases when the cardboard moves backward. However, due to frequency selective fading, CIR phase also exhibits different sensing qualities at different carrier frequencies. Comparing the Tap 1 phase values (upper panels) in Figs. 5a and 5b, we find that the moving object almost causes no impact to tap1 at f_{c1}, while phase exhibits clear increasing patterns in tap1 at f_{c2}. When applying fc2, all three taps are affected. This is because the multipath signals with corresponding delay similar to tap1~tap3 change when we move the cardboard forward and backward. The experiment results indicate that similar to the influence on CIR magnitude, frequency selective fading also affects the phase measurements at different frequencies.

Transmitting at multiple frequencies (e.g., OFDM) could enhance robustness against FSF since different frequency components are less likely to add up destructively at the same time. However, existing multi-frequency based methods incur high computational overhead due to Fast Fourier Transformation (*FFT*) and Inverse-FFT (*IFFT*) operations



(a) Frequency f_{c1}. (b) Frequency f_{c2}.

Fig. 5. CIR phase measurements of moving cardboard away from transceiver.



(a) Single-frequency. (b) Our frequency-hopping scheme.

Fig. 6. Different transmission schemes.

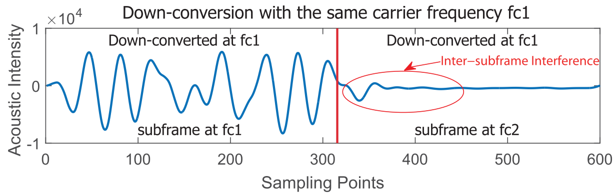
[22], [43]. In addition, OFDM-based method needs to add data-irrelevant Guarded Interval to remove ISI, which increases the time of a frame and decreases the time resolution for frame-based gesture recognition. Instead, we adopt frequency hopping to periodically transmit at different carrier frequencies to alleviate FSF. In particular, we transmit at a certain carrier frequency (e.g., f_{ci}) and hop to an adjacent frequency (e.g., f_{cj}). Thus, the whole channel measurement frame consists of N subframes transmitted at N different frequencies. Note that the frequency hopping scheme does not involve any *FFT* and *IFFT* operations, which reduces the computational overhead when extracting CIR measurements. Such a reduction of processing time is important especially when it is applied to resource-constrained smart devices.

However, due to sudden frequency transition from f_{ci} to f_{cj} at the edge of two adjacent subframes, the transmitted signal becomes audible to users. To keep the whole frame inaudible throughout the frequency hopping process, we apply a bandpass filter with passband $[f_{c1} - \frac{B}{2}, f_{cN} + \frac{B}{2}]$, which effectively filters out jitters at the edge of adjacent subframes, where B denotes the bandwidth. In practice, we append the first subframe and prepend the last subframe to a frame before passing through the bandpass filter. Then we remove the appended as well as the prepended subframes after applying the bandpass filter. The filtered inaudible frame can be saved as an audio file and played periodically at the transmitter.

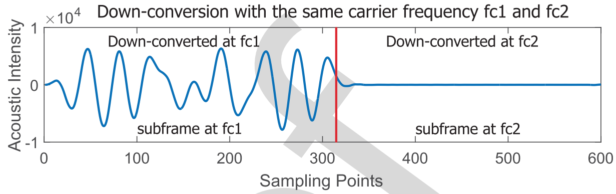
The receiver starts to record the reflected frame immediately after the first sample is emitted by the transmitter. To detect the position of the first sample in the received frame, we calculate the Pearson Correlation Coefficients (PCC) of the transmitted and the received audio samples and locate the peak of correlation. Once the first sample of the frame is detected, the boundary of subframes in the current frame and the subsequent frames can be easily located and perfectly synchronized due to fixed length of the subframe. Note that the frequency hops periodically from f_{c1} to f_{cN} within each received frame. The receiver down-converts the frame by multiplying each subframe with its corresponding $\cos(2\pi f_{ci} t)$ and $-\sin(2\pi f_{ci} t)$, where $i \in \{1, \ldots, N\}$ as shown in Fig. 6b. The down-converted frame then passes through a lowpass filter to filter out high-frequency components. Finally, the complex vector $r(t)$ of the same frequency are used for extracting CIR magnitude as well as CIR phase.

3.2.2 Remove Inter-Subframe Interference

Existing methods insert data-irrelevant cyclic prefix (i.e., multiple zeros) to avoid ISI, as shown in Fig. 6a. However,



(a) Received baseband frame.



(b) Received baseband frame without ISI.

Fig. 7. Remove the impacts of inter-subframe interference.

our down-conversion technique can naturally remove the ISI without inserting any prefix. To see how such a down-conversion technique avoids inter-subframe interference, we assume the current subframe is with frequency f_{cj}, which can be interfered by previous N subframes. Thus, the currently received subframe can be represented as $y(t) = \sum_{i=1}^{N} A_i \cos(2\pi f_{ci} t + \theta_i)$, where A_i is the amplitude of the subframes and θ_i is the phase offset caused by multipath effects, $i \in [1, N]$. By down-converting with $\cos(2\pi f_{cj} t)$, $j \in [1, N]$, we have

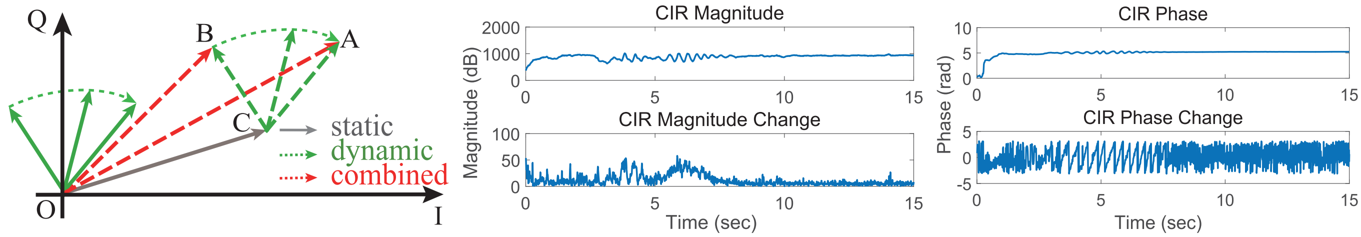
$$\begin{aligned}&\sum_{i=1}^{N} A_i \cos(2\pi f_{ci} t + \theta_i) \times cos(2\pi f_{cj} t)\\&= \sum_{i=1}^{N} \frac{A_i}{2} [(\underbrace{\cos(2\pi(f_{ci} + f_{cj})t + \theta_i)}_{\text{high-frequencycomponent}} + \underbrace{\cos(2\pi(f_{ci} - f_{cj})t + \theta_i))}_{\text{low-frequencycomponent}}].\end{aligned} \tag{1}$$

Looking at low-frequency component in Eq. (1), we have

$$\begin{aligned}&\sum_{i=1}^{N} \frac{A_i}{2} \cos(2\pi(f_{ci} - f_{cj})t + \theta_i)\\&= \frac{A_j}{2} \cos(\theta_j) + \sum_{i=1, i\neq j}^{N} \frac{A_i}{2} \cos(2\pi(f_{ci} - f_{cj})t + \theta_i).\end{aligned} \tag{2}$$

The high-frequency components in Eq. (1) and the second term in Eq. (2) can be simultaneously removed by applying a low-pass filter with a cutoff frequency set according to the difference of carrier frequencies (i.e., $min(|f_{ci} - f_{cj}|), i \neq j$). Besides, the cutoff frequency should exceed the frequency of the subframe such that the subframe can be recovered accurately. After passing the low-pass filter, we obtain $\frac{A_j}{2} \cos(\theta_j)$, where $\theta_j = \cos(2\pi f_{cj} \tau_j)$, and τ_j is the propagation delay. Since the speed of sound is known, with τ_j we can calculate the distance between the transceiver and the reflecting point.

To evaluate the effectiveness of our design, we conduct an experiment to compare ISI with/without our filtering method in Fig. 7. We transmit the first subframe at f_{c1}, followed by the second subframe at f_{c2}, and the carrier frequency hops at around the 320th sampling point. In the



(a) CIR phase and magnitude model. (b) CIR magnitude. (c) CIR phase.

Fig. 8. CIR phase and magnitude.

experiment, to better visualize ISI, the first subframe transmits TSC bits, while the second subframe contains zero samples, only to measure whether the first subframe would influence the second subframe. Fig. 7a shows the received frame down-converted with the same carrier frequency f_{c1} for both subframes. We see that the transmitted signal indeed echoed after the frequency hopping, which could have distorted the second subframe transmitted at the same f_{c1}. Fig. 7b plots the received signals when the first subframe is down-converted with frequency f_{c1}, while the second subframe with zero samples is down-converted with an adjacent frequency f_{c2}. We see that the first subframe transmitted at f_{c1} is correctly down-converted, and more importantly there is no interference or distortion in the second subframe. The experiment result shows that our filtering method can effectively remove Inter-symbol Interference.

3.2.3 Extract Effective CIR Phase and Magnitude

The extracted channel measurements involve both static objects in the environment (e.g., direct path from speaker to microphone, wall, desk, *etc.*) as well as dynamic objects (e.g., people passing by, *etc.*). Thus, the CIR measurements are the combinations of all signals reflected from both static and dynamic objects within the sensing range. To avoid the influence of static objects as well as moving objects irrelevant to the hand gesture, we need to extract the reflected signal from hands and fingers close to the transceiver.

Focus on Nearby Objects. In order to mitigate the influence of distant moving objects, we need to filter out the reflected signal from distant objects and only keep reflected signal from hands and fingers close to the transceiver. In the channel measurement, each tap of CIR corresponds to a certain delay range and reflected signals with similar propagation delays are grouped into one tap. Therefore, the tap index (e.g., Y-axis in Fig. 4) indicates the distance between the reflecting objects and the transceiver: the smaller the index, the closer to the transceiver. Thus, the detection range D_r can be set according to the number of taps L, since we have $D_r = L \times \frac{v}{2f_s}$, where v is the speed of sound and f_s is the sampling frequency. By tuning the detection range and only keeping a few effective taps, we can filter out the impact caused by objects outside a certain range to improve system robustness. This method ensures robust CIR measurement inside the detection range, even with people walking nearby but outside the detection range.

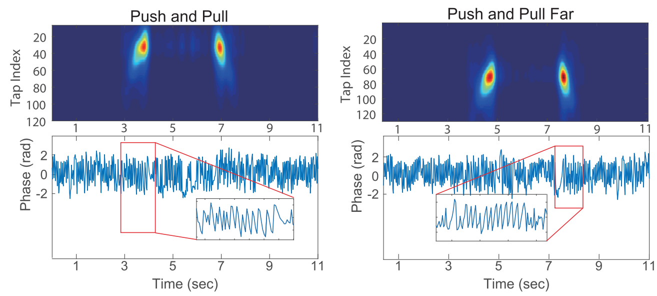
Focus on Moving Objects. The changes of combined phase and magnitude of CIR are illustrated in Fig. 8a. $\overrightarrow{OC}$ represents the static component with constant magnitude and phase, while $\overrightarrow{CA}$ and $\overrightarrow{CB}$ are the dynamic components with varying phases and magnitudes. The direct transmission from speaker to microphone and the static background reflection from the environment jointly comprise the static component. Due to the dynamic components, the combined components $\overrightarrow{OA}$ and $\overrightarrow{OB}$ change accordingly. Note that the CIR measurement only measures the combined components, while the static component and the dynamic component cannot be directly measured. To cancel the static component and extract the dynamic components from the measured CIR, we calculate the CIR difference between two consecutive measurements at time $t-1$ and t. In addition, the constant phase offset caused by the transmitter and receiver hardware can be removed as well by measuring the CIR differences. By doing this, the dynamic component can be extracted and the effects caused by surrounding static objects can be removed.

Figs. 8b and 8c show the CIR magnitude and phase of the same tap at the same carrier frequency extracted from the second experiment in Section 3.2.1. Due to the strong direct transmission from speaker to microphone, the pattern of original CIR magnitude and phase is not clear (upper panel in Figs. 8b and 8c). However, we observe that the extracted phase changes clearly exhibit linearly increasing patterns. Besides, we observe that CIR phase and magnitude vary differently since magnitude captures signal attenuation while phase captures propagation distance. Therefore, we may obtain more reliable information using both measurements.

3.3 Gesture Identifier

The main objective of the gesture identifier is to classify the CIR measurements and recognize different gestures. We notice that the CIR magnitude and phase across a certain time over multiple taps can be regarded as a CIR magnitude image and a CIR phase image, respectively. CIR images extracted from different frequencies can be considered as RBG channels. Recent advances in neural network and its breakthrough in image recognition motivate us to leverage such a powerful classification tool and build the gesture identifier. To this end, we weave the CIR measurements into tensors (named CIR images), which is similar to images in the context of image classification.

However, the neural networks require a huge amount of effective training data to achieve high accuracy and robustness. Ideally the training data should cover various practical scenarios. Yet, it takes a long time and a lot of effort to collect a sufficient amount of quality data in practice. To ease the pain of data collection, we conduct data augmentation



(a) Push and pull at $0 \sim 20$cm region. (b) Push and pull at $20 \sim 40$cm.

Fig. 9. CIR phase and magnitude of push and pull.

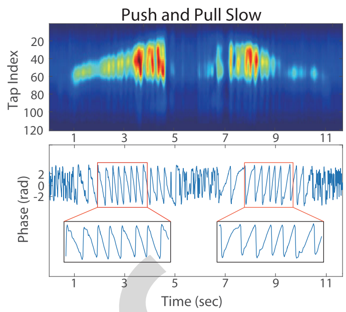


Fig. 10. Push and pull at slow speed.

to enrich our training data so that the augmented data can reflect different variations of CIR measurements without manually collecting the data in all possible scenarios.

3.3.1 Impact Factor Investigation

The data augmentation technique relies on our key observation that the CIR measurements vary along with the gesture variations (e.g., gesture speeds, angles, positions and etc.). Based on our initial measurement results, we mainly consider five factors that could affect the CIR data in real practical impact factors including gesture speed, distance to microphone, angle of arrival, blockage of line-of-sight path, and background noise. We then apply data augmentation techniques that are widely used in image processing [9], [39], [50] on original CIR data (e.g., translation and scaling) so that the augmented CIR data can cover potential scenarios and the trained models can cope with the above influential factors.

Different Distances to the Receiver. In commodity smartphones, the speaker and microphone are typically collocated and built into a single device. To measure the influence of the distance between a hand and the transceiver, we perform push and pull at a distance between hand and transceiver ranging from 0cm to 20cm, and then 20cm to 40cm in front of the transceiver, respectively. Figs. 9a and 9b show the CIR magnitude (upper panel) and phase measurements (lower panel), respectively.

Comparing Figs. 9a and 9b (upper panel), we observe vertical drift in tap indexes in CIR magnitude measurements. That is because the gestures are performed at different distances to the transceiver. A larger tap index indicates a further distance to the transceiver. Similarly, we find corresponding shifts in CIR phase measurements. As illustrated in Figs. 9a and 9b (lower panel), we observe similar linearly increasing patterns in CIR phase measurements. Therefore, CIR measurements of gestures performed at different distances to the smartphone can be emulated by vertical drifts in tap indexes within the sensing range of the receiver.

Different Speeds. To illustrate the impact of different moving speeds of gestures, we perform push and pull at a relatively slow speed in front of the transceiver within 20cm. Fig. 10 shows the CIR magnitude for all taps and CIR phase for one particular tap. The CIR phase rotation indicates the path length change caused by the moving hand. The key observation is that the CIR measurements corresponding to the gesture expand in time in both CIR magnitude and phase compared to Fig. 9a due to the slower speed. To compensate for different speeds of gestures, we perform data augmentation by horizontally expanding or contracting an original CIR measurement to emulate different speeds. In our work, a gesture takes at least 0.4s and each frame lasts for 19.5ms (6.5ms/subframe for 3 subframes). Around 20 frames are received in 0.4s for each frequency to estimate the CIR. We notice that, when less than 20 frames are used for CIR measurement, the gesture may not be correctly identified.

Blockage of Transceiver. People may attempt to control their smart devices under NLOS case. To simulate this scenario, we place a smartphone inside a cotton bag to capture the moving hand. In upper panel of Fig. 11, we observe less bright patterns if we directly use raw CIR data. In practice, NLOS may cause signal attenuation, which results in very small values of CIR magnitude.

To address this problem, we use the Min-Max Normalization method to scale and normalize the CIR magnitude measurements. After normalization, all the magnitude values are scaled to the same level (i.e., $0 \sim 1$) such that the impact of signal attenuation can be mitigated. The lower panel in Fig. 11 shows the normalized CIR measurements of the raw CIR data in the upper panel. After normalization, we observe similar patterns compared to the scenario without any blockage in Fig. 9a. We observe consistent patterns when we place a thick paper between transceiver and hand. On the contrary, the CIR phase measurements are not greatly affected due to similar relative moving distances of hand. In all experiments, we conduct normalization to all raw CIR data before data augmentation.

Noisy Environment. To evaluate the impact of background noise, during CIR measurement, we use a smartphone to play music 5cm away from the receiver. In this case, the

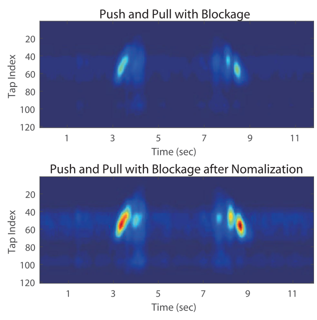


Fig. 11. Push and pull with blockage.

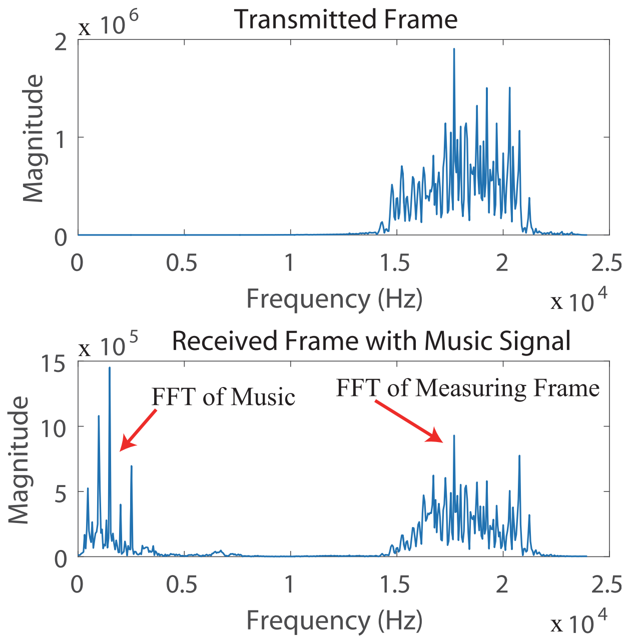


Fig. 12. Frequencies of the Transmitted and Received frame.

received signal is a mixed signal of both TSC signal and the background music signal. Fig. 12 (upper panel) shows the frequencies of the transmitted TSC signal and Fig. 12 (lower panel) shows the received mixed signal, respectively. In the figure, we see that the music resides in the frequency band much lower than the transmitted inaudible signal. As such, the receiver can separate the transmitted inaudible signal from the background noise in the environment (e.g., music) in the frequency domain.

Intuitively, we can add a high-pass filter before down-conversion to remove the low frequency components. In fact, our down-conversion and demodulation method (described in Section 3.2.1) can filter the music and other noises in the low frequency band. Suppose the highest frequency component in music is $A_m \cos(2\pi f_m t)$, where A_m and f_m denote the corresponding amplitude and frequency. In down-conversion step, we have

$$\begin{aligned} & A_m \cos(2\pi f_m t) \times \cos(2\pi f_c t) \\ & = \frac{A_m}{2}\left[\cos(2\pi(f_m + f_c)t) + \cos(2\pi(f_c - f_m)t)\right], \end{aligned} \tag{3}$$

where f_c is the corresponding carrier frequency. We notice that the frequencies of most music signals are lower than 8 KHz. In contrast, the TSC is transmitted at much higher frequencies over 16 KHz. Hence, the frequency components $f_m + f_c$ and $f_m - f_c$ can be filtered out when $f_m < f_c - \frac{B}{2}$.

Actually, many other background noises (e.g., human voice, fans, air conditioner, traffic noise, *etc.*) reside in low-frequency bands, which can be similarly filtered out by our down-conversion and demodulation method. Therefore, there is no need to add a high-pass filter before down-conversion. In other words, the down-conversion and demodulation method is inherently robust against background noises.

Different Angles. In order to evaluate the impact of angle-of-arrival on the transceiver, we perform gestures around the transceiver at different angles within 20 cm range to the transceiver. In particular, we divide the $0^\circ \sim 180^\circ$ area in front of the transceiver into three 60° sectors (i.e., $0^\circ \sim 60^\circ$, $60^\circ \sim 120^\circ$, and $120^\circ \sim 180^\circ$) and perform push and pull multiple times in each sector. The experiment results show that the CIR measurements exhibit similar patterns when we perform the same gesture from different angles ($0^\circ \sim 60^\circ$, and $120^\circ \sim 180^\circ$) as in Fig. 9a ($60^\circ \sim 120^\circ$). This is because both speaker and microphone are omnidirectional. In fact, omnidirectional speakers and microphones are widely used in commodity smart devices in order to achieve good quality in all directions. Besides, the speaker and the microphone are collocated in a single device with short distance. As such, the impact of angle-of-arrival on the CIR measurement is limited. Thus, in this work, we do not augment the raw measurements for different angle-of-arrivals.

In summary, we find that the last three factors (i.e., blockage, noise and angle-of-arrival) do not require any particular data augmentation, while different speeds and distances to the receiver do influence the CIR measurements and need careful treatment. Note that different hand sizes of users may influence the CIR measurements. However, with multiple taps, our method can reduce the impact of hand sizes.

We assume that the gestures are performed while the user is standing or sitting still with static torso but only moving his hand. In practice, people often perform gestures at distance $10 \sim 50$cm to the transceiver, which indicates tap indexes ranging from 30 to 150. We guarantee the successful transmission and reception of the audio signal within this detection range. Thus, we vertically shift a raw CIR data according to the targeted tap index ranges. One may freely adapt the tap index range according to different practical impact factors by tuning appropriate volume of speaker if the distance between hand and transceiver increases. On the other hand, we find that the largest difference between the speeds for the same gesture is typically at most $5\times$ (i.e., 0.4s to 2s). As such, the number of horizontal expanding and contacting rates are varied from 2 to 5. Although the largest speed difference in our dataset is up to $5\times$, the data augmentation technique is not limited to this range and can be extended to a larger range to emulate more variances in practice (e.g., $4s$ for push in Fig. 10). We randomly combine the above settings for various gesture speeds and distances and augment $100\times$ for each collected gesture to emulate the gestures performed under various practical scenarios.

3.3.2 *Gesture Recognition*

We input the augmented training CIR data into a classifier to identify different gestures. Recently, CNN exhibits significant advances in image recognition while LSTM is promising to process time series data. Therefore, our classifier consists of a CNN for extracting significant features of CIR images and an LSTM network for gesture identification.

In specific, we separately process CIR magnitude and phase and automatically extract features with two independent CNNs but with the same architectures. We apply a CNN with five convolution layers. Each input of the first convolution layer is a CIR image with size $[K \times L \times N]$, where L is the number of taps, K denotes the number of consecutive subframes aggregated during a certain period and N is the number of frequencies. Note that similar to the real images, CIR images extracted from different frequencies can be regarded as different image channels (e.g., RGB channels). We use 32 kernels with size $[5 \times 5 \times N]$ to scan the input image, followed by a max-pooling layer with $[2 \times 2]$ kernel and stride length 2. The design of the remaining 4 convolution layers are similar to the first layer with one kernel size $[5 \times 5]$ and three kernel sizes $[3 \times 3]$, and the number of kernels are set to two 32 and two 64, respectively. The activation function is ReLU. We set a fully connected layer with size 512 to output the feature vector. The extracted features of CIR magnitude and phase are then processed separately with two individual LSTM.

When performing different gestures (e.g., up and down, left and right), the same feature extracted with CNNs may appear in different order and the order matters in distinguishing the different gestures. Unlike CNN, LSTM is capable of memorizing the context information in sequential data [10], which can capture the temporal information of the gestures. In our implementation, the LSTM architecture takes multiple outputs of the CNN across time into one vector as the input data. We use one stacked LSTM layer grouped by 8 memory cells. A softmax function layer is used after the LSTM layer to predict the gesture types. The output of the LSTM is a probability vector indicating the likelihood of different gestures. Note that, we separately build two LSTMs for CIR magnitude and phase image and generate two probability vectors. The gesture type is then determined by the equally weighted sum of the two probability vectors.

4 Experiment and Evaluation

4.1 Experiment Setting

Parameter Setting. To transmit channel measurement frame with frequency hopping, frequencies that satisfy with conditions in Section 3.2.2 can be applied to mitigate the frequency selective fading and remove inter-subframe interference. In our experiment, RobuCIR emits inaudible signals at three frequencies 18 KHz, 20 KHz and 22 KHz, respectively. We notice that the acoustic signals played at the maximum volume may still be noticed by some users, especially when they really pay attention in quiet rooms. Users can adjust the volume to their comfortable level (e.g., 75 percent of maximum volume) without affecting much the system performance.

In our design, we choose a 26-bit TSC, which has excellent autocorrelation and synchronization property [28]. The up-sampling rate is set to 12. Therefore, a single TSC symbol is represented by 12 audio samples and each transmitted subframe contains $N_{TSC} \times 12 = 312$ audio samples, which takes 6.5ms in transmission with sampling rate of 48 KHz.

Data Collection. We implement RobuCIR on a Samsung S9 Plus, a Samsung S7 Edge and a Google NEXUS5 phone. Experiment results show that the diversity of smartphones (e.g., signal distortion at high frequencies) can be mitigated by frequency hopping, normalization, and data augmentation. We invite 8 volunteers (5 males and 3 females) to perform 15 types of gestures. Each gesture is repeated 6 times (3 for each hand) under 5 practical impact factors described in Section 3.3. The users stand or sit still at 0.5m to 1m from the device and perform gestures with relatively static torso and move their hands within the detection range of up to 0.5m. Because it is very hard to measure the exact speed of a gesture, instead, we use the time duration of the gesture to represent different speeds of gestures. The largest speed difference in our dataset is $5\times$ (e.g., from 0.4s to 2s) and a gesture with faster speed has shorter duration, and vice versa. We place the test smartphone into a cotton bag to emulate the NLOS scenario. The gestures are performed at different angles to the device ranging from $0^\circ \sim 180^\circ$ within 20cm range to the transceiver. In particular, we divide the $0^\circ \sim 180^\circ$ area in front of the transceiver into three sector with the same angle. Performing gestures at different sectors results in the received signal arrived in different angles. In the noisy environment scenario, we use another mobile phone as an external speaker to play music with the largest volume placed 0.5m away from the target device. The gestures are performed at different time and different environments containing some rich multipath office rooms between size $10 \times 8 \times 3\text{m}^3$ and $4 \times 4 \times 3\text{m}^3$ with different layouts. These office rooms are surrounded by furniture, computers and small objects nearby, which result in different signal decay. People are allowed to move near the target device when we are collecting the data. In total, we collect 3600 real gesture samples.

Benchmark. We evaluate the performance in comparison with the state-of-the-art UltraGesture [17] as our benchmark. UltraGesture is configured and optimized according to [17] to achieve its best performance. We set the same number of estimated taps to $L = 140$ in magnitude measurements. We choose $K = 32$ and $N_{lstm} = 5$ such that the LSTM takes features of $K \times N_{lstm} \times 6.5ms \approx 1sec$ as each input.

Model Training and Gesture Recognition. We use 10-fold cross-validation to evaluate the robustness of the system. Each round of cross-validation involves training a new model with the collected samples from 6 users and testing with the collected samples from the other 2 users. We make sure that the training data and the testing data are collected from different users and different rooms in each round. For each gesture in the training group, we conduct data augmentation with rate $= 100\times$. We notice that the augmented samples are consistent with the corresponding real-world scenarios.

The classifier are trained using TensorFlow in a high-end server with Intel(R) Xeon(R) E5-2620 v4 CPU @2.10 GHz, 32 GB memory, and two Nvidia GTX 1080 Ti GPU graphics cards. It takes around 65s for each training iteration. Note that the model training is a one-off procedure and can be carried out offline. The size of the model when using 5-layer CNN and 8-cell 1-layer LSTM is around 5.5M. We use the high-end server with the same specifications to simulate a cloud/edge server and conduct performance evaluation.

4.2 Evaluation

4.2.1 *Overall System Performance*

Fig. 13 shows the overall confusion matrix of our RobuCIR system for all 15 gestures performed at different rooms with different environments. Some rooms are with rich multipath, which are surrounded by furniture, computers and small

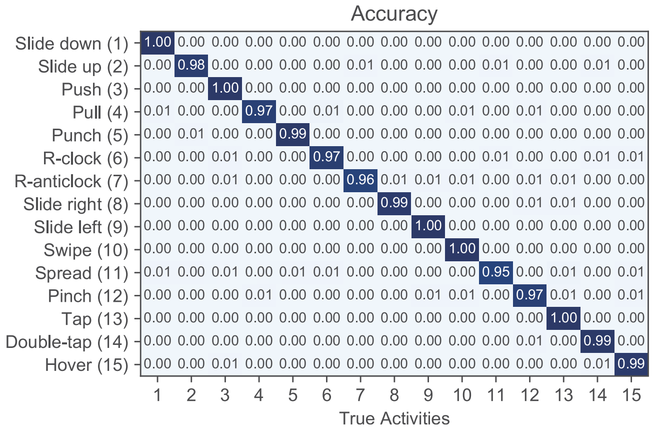


Fig. 13. Overall performance of RobuCIR.

objects nearby, while some rooms are relatively empty with less multipath. The test data was collected at different distances to the transceiver and the volunteers perform the gestures at their comfortable speeds in office rooms. RobuCIR achieves an average recognition accuracy of 98.4 percent, and each gesture exceeds 95 percent accuracy even under different practical impact factors. Different environments with different signal fading have limited impact on system performance, since the detection range can be set with the number of CIR taps to filter out interference and multipath reflection outside the detection range (e.g., people walking around).

We evaluate the recognition accuracy under different practical impact factors, as shown in Fig. 14. The accuracy of all gestures exceeds 96 percent, which demonstrates high robustness of RobuCIR under various scenarios. The accuracy when performing gesture at different speeds and different distances to the transceiver is slightly lower than other three scenarios since these two scenarios may cause larger variations in CIR measurements while other three scenarios do not introduce dramatic influence in CIR measurements.

4.2.2 Improvement of Robustness

To evaluate system robustness of RobuCIR compared to the existing works, we compare the performance with the state-of-the-art work UltraGesture [17] which is trained and evaluated with the same dataset. We set the same parameters as presented in UltraGesture and evaluate both RobuCIR and UltraGesture under various practical impact factors. In our experiment, we use 10-fold cross validation and take the

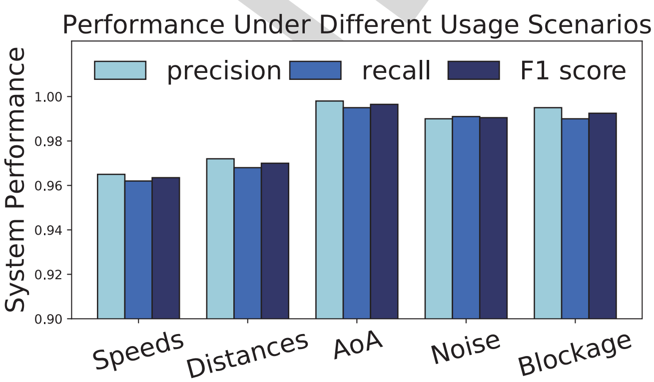


Fig. 14. Performance with different practical impact factors.

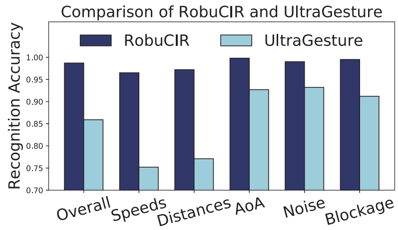


Fig. 15. Results of RobuCIR and UltraGesture.

average accuracy, which is compared to the UltraGesture. The standard deviation of 10-fold cross validation is less than 1.4 percent with the lowest and highest accuracy of 96.9 and 100 percent, respectively. Fig. 15 shows the recognition accuracy of RobuCIR and UltraGesture.

As illustrated in Fig. 15, RobuCIR substantially outperforms UltraGesture and achieves overall recognition accuracy of 13 percent higher than UltraGesture. When performing gestures at different speeds and different distances to the transceiver, RobuCIR remains robust with an accuracy of over 96 percent, while the performance of UltraGesture dramatically decreases to 75 and 77 percent mainly due to FSF and considerable impacts on CIR measurements under those two scenarios. For other three practical impact factors, the performance of UltraGesture exceeds 90 percent while RobuCIR achieves higher accuracy of over 98 percent since the augmented training data covers different variations of gestures under practical scenarios.

4.2.3 Impact of Frequency-Hopping

To evaluate the frequency hopping scheme, we evaluate RobuCIR with different single-frequency signals. In this experiment, we separately train three neural networks according to different frequencies. To focus on the impact of frequency-hopping scheme, we keep all the parameters unchanged. Fig. 16 illustrates the recognition accuracy of RobuCIR under different practical impact factors evaluated using three single-frequency signals.

We observe that the performance of RobuCIR varies under the same practical impact factors when transmitting different single-frequency signals. When only transmitting signal with frequency2, the performance decreases significantly to 81 and 78.2 percent under different speeds and distances to transceiver scenarios since the measured signal might be destructively added up when a hand is at a specific location. As such, the extracted CIR measurements fail to reflect the patterns of corresponding gestures. In contrast, when applying frequency-hopping scheme, we can simultaneously acquire consistent CIR measurements derived from other frequencies (i.e., frequency1 and frequency3). Therefore, more effective

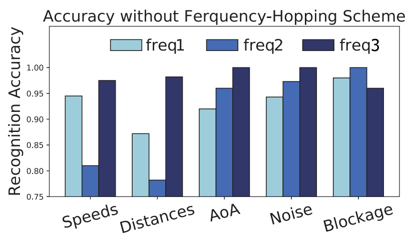


Fig. 16. Accuracy without frequency-hopping.

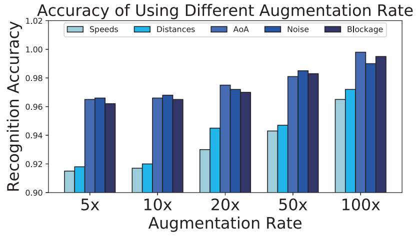


Fig. 17. Accuracy with different rate.

features can be extracted by the neural networks, which enhances the system robustness.

4.2.4 Impact of Data Augmentation

We vary the data augmentation rates (i.e., $5\times \sim 100\times$) and train classifier with different augmented data. In this experiment, we transmit TSC using frequency-hopping scheme with three carrier frequencies, and other parameters remain the same.

The results in Fig. 17 show that the recognition accuracy of RobuCIR under all scenarios improves as the augmentation rate increases. In particular, the accuracy when performing gesture under different speeds and distances experiences higher increase than other three scenarios since data augmentation is carefully applied under these two scenarios and a larger augmentation rate covers more variations of the gesture. As the augmentation rate raises to $100\times$, the accuracy for each scenario exceeds 96 percent. The experiment results demonstrate that the data augmentation techniques indeed provide more insights and quality data to the neural networks and help improve the system robustness.

4.2.5 Impact of Neural Network Settings

1) Impact of CNN architecture: To evaluate the impact caused by the CNN settings and its efficacy in extracting useful features, we vary the number of CNN layers from 2 to 7 while keeping the LSTM architecture unchanged. We transmit the signal with frequency hopping scheme and augment the training data $100\times$. For each network, we set the first layer with $[5\times 5]$ kernel and the rest layers with $[3\times 3]$ kernel. A max-pooling layer with $[2\times 2]$ kernel and stride length 2 is added after each layer. The number of kernels for the first two layers is 64 and the rest is 32. During CNN training stage, we notice that the 5-layer CNN generally start to converge after 100 iterations for an augmented training dataset of 180000 samples. Therefore, we set the number of iterations to 100 when training models with different number of convolution layers.

As depicted in Table 1, we observe that using more number of convolution layers achieves better performance. We have tested CNN with a number of layers larger than 5 and find not much improvement in performance. Therefore, we choose 5-layer CNN for extracting the features.

2) Impact of LSTM architecture: In this experiment, we vary the number of LSTM cells from 2 to 8 while keeping the number of CNN layers to 5 and other experiment settings unchanged. The results show that with the number of cells in LSTM layer increases, the system performance improves correspondingly, as in Table 2. However, the marginal gain

TABLE 1
Performance With Varied # of CNN Layers

# of layers	2	3	4	5	6	7
precision	0.94	0.95	0.96	0.99	0.99	0.99
recall	0.93	0.94	0.95	0.98	0.99	0.99
F_1 score	0.93	0.95	0.95	0.98	0.99	0.99

$F_1\ score = 2\times\frac{precision\times recall}{precision+recall}$

of further increasing the number of cells in LSTM layer beyond 8 is small. As such, the number of cells in our LSTM layer is set to 8.

4.2.6 Execution Time

We run 20000 inferences and measure the average execution time. Table 3 shows the execution time of RobuCIR at each processing stage. Frame detection by calculating correlation coefficient is performed every time before a gesture and down-conversion step is needed throughout the CIR measurement processing stage, which take approximately 1.3 ms and 2.2 ms, respectively. LS estimation for generating CIR magnitude and phase takes a bit longer time of 4.8ms depending on the number of configured taps. Our trained deep learning model can process each CIR measurement within an average of 23 ms at the high-end server. As a result, the execution time of RobuCIR is approximately 31 ms. We note that the acoustic data needs to be offloaded to a cloud/edge server for processing, which involves extra round-trip time depending on network conditions.

Our current implementation of RobuCIR primarily focuses on enhancing the robustness of the acoustic sensing performance. To reduce the computational overhead at the mobile device side, we offload the computation-intensive task involved in gesture recognition to the high-end server. With this design consideration, we expect to support lightweight resource-constrained smart devices (e.g., smart speaker, smart watch), which cannot immediately afford the computational overhead at this moment. In our experiment, we use smartphone to emit and receive the acoustic signal. The received acoustic signal is saved as a file in the smartphone. The file is wirelessly transmitted to the high-end PC using file transferring APP via WiFi. We notice that many wired technologies can be used to transfer the file from the device to the server such as 5G, WiFi, Bluetooth and etc. In our case, we ignore the transferring time because the file can be transferred to server in real-time once it has been created if under good network conditions. However, such offloading manner will introduce extra delays under bad network condition.

Recent advances in running deep neural network models on mobile devices have achieved remarkable results through

TABLE 2
Performance With Varied # of LSTM Cells

# of cells	2	4	6	8	10
precision	0.91	0.93	0.98	0.99	0.99
recall	0.90	0.92	0.97	0.98	0.98
F_1 score	0.90	0.92	0.97	0.98	0.98

$F_1\ score = 2\times\frac{precision\times recall}{precision+recall}$

TABLE 3
The Running Time of RobuCIR

CIR measurements Calculation			Gesture Recognition
Frame detection	Down-conversion	LS	Coupled NN model
1.3ms	2.2ms	4.8ms	23ms

model compression, cloud-free DSP, system optimization, *etc*. [5], [6], [8], [12], [16], [46], [50]. DeepASL [5] designs a transformative deep learning-based sign language translation technique and applies the trained neural network to the devices with processing latency in ms-level. NestDNN [6] enables resource-aware multi-tenant on-device deep learning by dynamically selecting the optimal resource-accuracy trade-offs, which is applied to the mobile devices with limited resources. DeepMon [12] employs a VGG-VeryDeep-16 deep learning model on smartphones by applying a suite of optimization techniques and can classify an image within a second. To avoid the extra latency involved in the network, one may embed the trained model and directly run on smartphones or even lightweight smart devices by leveraging the latest development of mobile computing. For example, Tensorflow Lite [38] can be used to run machine learning models on mobile and embedded devices with low latency. We plan to study this problem for future work.

5 Discussion

Privacy. As we use speakers and microphones to measure CIR data and need to offload to a cloud/edge server to process the CIR data, users may be concerned whether such CIR data would leak private information (e.g., private conversation). As a matter of fact, the CIR is measured in the high frequency band (e.g., ≥ 18 KHz), and only the preprocessed data will be offloaded to the server. It means that no conversation will be transmitted to the server.

Power Consumption. Current version of RobuCIR has not yet been extensively optimized for energy efficiency. In working mode, it needs to constantly transmit and receive acoustic signals to measure CIR, which incurs relatively high power consumption. Such power consumption is acceptable for smart speakers at home or in car, but cannot be afforded by mobile devices with limited battery life (e.g., smart watch). To reduce the power consumption in practice, a low-power component (e.g., IMU, light sensor) can be used to trigger and wake up RobuCIR in idle mode.

Motion Artifacts. In our current work, we assume the user's torso and the device are relatively static such that only the movement of hand is captured by the transceiver. In practice, the mobile phone and human torso might be in dynamic status (i.e., walking with mobile phone in the pocket), which results in inconsistent hand moving distance, speed and AoA. Besides hands' location relative to the transceiver (e.g., AoA), hand orientation when performing gestures may cause different CIR measurements as well. Such relative motions between the user's hand and the mobile device could affect the performance of our system. We plan to address these practical challenges in the future.

Model Sizes. As we apply neural networks to identify the gesture types, there exists tradeoff between the system performance and the model size of the neural network. A deeper neural network achieves higher performance while inevitably resulting in larger model size, and vice versa. Our neural network is 5 layers of CNN and 1 layer of LSTM and the current model size is 5.5M. We notice that although model size is not a problem for high-end servers, it cannot be ignored if applied to the resource-constrained smart devices. A larger model size gives rise to higher RAM and increases the processing time of identifying a single gesture, which costs higher power consumption for smart devices. Models with smaller sizes are more appropriate for resource-constrained smart devices while with lower system performance. One possible approach is to deploy the trained model on the cloud and only extracting CIR measurement at the end devices. The CIR measurement is sent to the cloud for gesture identification once it is measured by the smart devices and the cloud sends back the results.

6 Related Work

In recent years, contact-free gesture recognition techniques enable human-computer interaction. They realize control of machine by performing gestures nearby the devices without any contact. Camera-based gesture recognition system has been embedded in current vehicles (e.g., BMW) and smart home systems, which allow users to control speaker volume while chatting in the car or control smart devices at home. However camera-based systems rely on LoS path and good lighting conditions, which limit its practical impact factors. Google Soli uses a specialized radar to transmit millimeter waves to control the devices, which has been integrated into latest smartphones (iPhone & Google Pixel). However, it works in the 60 GHz frequency range, which is used for special purposes and may not be allowed in some countries. FMCW radar and USRP have been used to track human gestures [14], [27]. However, they require specialized devices and incur high deployment cost. RobuCIR exploits widely used speaker and microphone to transmit and receive acoustic signals, which works under 18 KHz to 24 KHz frequency band and does not rely on LoS path and lighting conditions.

As speakers and microphones are widely deployed in various smart devices (e.g., smartphone, smart speaker, smart watch), acoustic sensing has attracted wide attention in both industry and academia [3], [7], [15], [17], [19], [21], [22], [23], [24], [25], [30], [31], [33], [34], [40], [42], [43], [47], [49], [51], [52], [53]. SoundWave [7] can detect gestures by tracking hand motion (e.g., speed, direction, and amplitude) based on the Doppler shift of the audio signals reflected from the hands. AudioGest [30] can identify six types of gestures with high accuracy by measuring Doppler shift. EchoTrack [3] recognizes gestures based on the Time-of-Flight information. FingerIO [22] measures the change in the cross-correlation of the consecutive received acoustic signals to track the moving hand. However, FingerIO treats the whole hand as a single reflection point to track the hand movement, which cannot capture the complex finger movement of gestures. Our RobuCIR can effectively measure the multipath reflection from fingers when performing gestures by applying CIR. LLAP [43] enables trajectory tracking of a finger by extracting signal phase information. Strata [49] achieves higher accuracy by measuring CIR of the reflected audio signals. However, Strata still regards the finger as a signal reflection point. In our work, we apply both CIR magnitude and phase to measure the signal reflection, which provide different yet effective information of gestures. Those works regard the finger/hand as a single reflection point and achieve high tracking accuracy. However, modeling the whole hand as a single point fails to provide sufficient resolution. UltraGesture [17] measures CIR magnitude of the reflected audio signal and recognizes hand gestures. However, UltraGesture suffers from frequency selective fading and needs a huge amount of training data to effectively train neural network models. Unlike UltraGesture that emits single frequency signal, we exploit frequency hopping scheme to mitigate frequency selective fading. Besides, to obtain sufficient training data and increase system robustness, we apply the data augmentation technique to automatically generate training data. In summary, unlike these works, we present a holistic design and implementation of robust CIR measurement, data augmentation, and learning based classification, which as a whole improves the overall performance in terms of accuracy and robustness.

Radio frequency (RF) signals are used to track finger/hand motion [1], [4], [11], [14], [26], [27], [35], [36], [41], [45]. AllSee [14] recognizes gestures using power-harvesting sensors. RfIDraw [41] and RFIPad [4] track the trajectory of finger movement and enable in-air handwriting. WiGest [1] leverages WiFi signal strength to recognize gestures near mobile devices. WiSee [27] can track different home gestures by extracting minute Doppler shifts of WiFi signals induced by human body. WiFinger [36] can recognize gestures by detecting unique patterns in Channel State Information (CSI). WiDraw [35] enables hands-free in-air drawing by processing the Angle-of-Arrival values of incoming WiFi signals. Such works require RF devices and support different applications from acoustic based works.

Vision based gesture tracking are well-studied [20], [29], [32]. Microsoft HoloLens [20] uses specialized cameras to provide contact-free human gesture tracking. Sony PlayStation VR [32] require users to wear helmets and controllers, which are cumbersome compared to contact-free systems. DigitEyes [29] can model hand movement from ordinary gray-scale images. However, vision based methods require good light conditions, which limits their applications.

7 Conclusion

This paper presents a holistic design and implementation of an acoustic based gesture recognition system that can identify 15 types of gestures with high robustness and accuracy. In order to alleviate frequency selective fading, this paper adopts frequency hopping and carefully designs down-conversion and demodulation to avoid inter-subframe interference. Based on the insights obtained in the initial experiments, this paper conducts data augmentation on raw CIR data to synthesize new augmented data, which is used to effectively train neural network models. In particular, the augmented data captures different variations in practical scenarios such as different gesture speeds, distances to transceiver, and signal attenuation. The experiment results show that RobuCIR substantially outperforms state-of-the-art work and achieves an overall accuracy of 98.4 percent under different practical impact factors.

Acknowledgments

The authors would like to thank the editor and reviewers for their help and insightful comments. This work was supported in part by the National Nature Science Foundation of China under grant 61702437 and Hong Kong GRF under Grant PolyU 152165/19E.

References

[1] H. Abdelnasser, M. Youssef, and K. A. Harras, "WiGest: A ubiquitous WiFi-based gesture recognition system," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 1472–1480.

[2] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3D tracking via body radio reflections," in *Proc. 11th USENIX Conf. Netw. Syst. Des. Implementation*, 2014, pp. 317–329.

[3] H. Chen, F. Li, and Y. Wang, "EchoTrack: Acoustic device-free hand tracking on smart phones," in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.

[4] H. Ding *et al.*, "RFIPad: Enabling cost-efficient and device-free in-air handwriting using passive tags," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 447–457.

[5] B. Fang, J. Co, and M. Zhang, "DeepASL: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation," in *Proc. 15th ACM Conf. Embedded Netw. Sensor Syst.*, 2017, Art. no. 5.

[6] B. Fang, X. Zeng, and M. Zhang, "NestDNN: Resource-aware multi-tenant on-device deep learning for continuous mobile vision," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 115–127.

[7] S. Gupta, D. Morris, S. Patel, and D. Tan, "SoundWave: Using the doppler effect to sense gestures," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2012, pp. 1911–1914.

[8] S. Han, H. Shen, M. Philipose, S. Agarwal, A. Wolman, and A. Krishnamurthy, "MCDNN: An approximation-based execution framework for deep stream processing under resource constraints," in *Proc. 14th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2016, pp. 123–136.

[9] S. Hauberg, O. Freifeld, A. B. L. Larsen, J. Fisher, and L. Hansen, "Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation," in *Proc. 19th Int. Conf. Artif. Intell. Statist.*, 2016, pp. 342–350.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] Y. Hou, Y. Wang, and Y. Zheng, "TagBreathe: Monitor breathing with commodity RFID systems," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 404–413.

[12] L. N. Huynh, Y. Lee, and R. K. Balan, "DeepMon: Mobile GPU-based deep learning framework for continuous vision applications," in *Proc. 15th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2017, pp. 82–95.

[13] W. Jiang *et al.*, "Towards environment independent device free human activity recognition," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 289–304.

[14] B. Kellogg, V. Talla, and S. Gollakota, "Bringing gesture recognition to all devices," in *Proc. 11th USENIX Conf. Netw. Syst. Des. Implementation*, 2014, pp. 303–316.

[15] H. Khan, U. Hengartner, and D. Vogel, "Augmented reality-based mimicry attacks on behaviour-based smartphone authentication," in *Proc. 16th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2018, pp. 41–53.

[16] N. D. Lane *et al.*, "DeepX: A software accelerator for low-power deep learning inference on mobile devices," in *Proc. 15th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, 2016, pp. 1–12.

[17] K. Ling, H. Dai, Y. Liu, and A. X. Liu, "UltraGesture: Fine-grained gesture sensing and recognition," in *Proc. 15th Annu. IEEE Int. Conf. Sens. Commun. Netw.*, 2018, pp. 1–9.

[18] W. Mao, J. He, and L. Qiu, "CAT: High-precision acoustic motion tracking," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 69–81.

[19] W. Mao, M. Wang, and L. Qiu, "AIM: Acoustic imaging on a mobile," in *Proc. 16th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2018, pp. 468–481.

[20] Microsoft, "Hololens," 2018. [Online]. Available: https://www.microsoft.com

[21] R. Nandakumar, S. Gollakota, and N. Watson, "Contactless sleep apnea detection on smartphones," in *Proc. 13th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2015, pp. 45–57.

[22] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "FingerIO: Using active sonar for fine-grained finger tracking," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2016, pp. 1515–1525.

[23] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "BeepBeep: A high accuracy acoustic ranging system using cots mobile devices," in *Proc. 5th Int. Conf. Embedded Netw. Sensor Syst.*, 2007, pp. 1–14.

[24] C. R. Pittman and J. J. LaViola, "Multiwave: Complex hand gesture recognition using the doppler effect," in *Proc. 43rd Graph. Interface Conf.*, 2017, pp. 97–106.

[25] S. Pradhan, G. Baig, W. Mao, L. Qiu, G. Chen, and B. Yang, "Smartphone-based acoustic indoor space mapping," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 2, pp. 75:1–75:26, Jul. 2018. [Online]. Available: http://doi.acm.org/10.1145/3214278

[26] S. Pradhan, E. Chai, K. Sundaresan, L. Qiu, M. A. Khojastepour, and S. Rangarajan, "RIO: A pervasive RFID-based touch gesture interface," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, 2017, pp. 261–274.

[27] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Netw.*, 2013, pp. 27–38.

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

[29] J. M. Rehg and T. Kanade, "Visual tracking of high DOF articulated structures: An application to human hand tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 35–46.

[30] W. Ruan, Q. Z. Sheng, L. Yang, T. Gu, P. Xu, and L. Shangguan, "AudioGest: Enabling fine-grained hand gesture detection by decoding echo signal," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 474–485.

[31] J. Shen, O. Lederman, J. Cao, F. Berg, S. Tang, and A. Pentland, "GINA: Group gender identification using privacy-sensitive audio data," in *Proc. IEEE Int. Conf. Data Mining*, 2018, pp. 457–466.

[32] Sony, "PlayStation VR," 2018. [Online]. Available: https://www.playstation.com

[33] K. Sun, W. Wang, A. X. Liu, and H. Dai, "Depth aware finger tapping on virtual displays," in *Proc. 16th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2018, pp. 283–295.

[34] K. Sun, T. Zhao, W. Wang, and L. Xie, "VSkin: Sensing touch gestures on surfaces of mobile devices using acoustic signals," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 591–605.

[35] L. Sun, S. Sen, D. Koutsonikolas, and K.-H. Kim, "WiDraw: Enabling hands-free drawing in the air on commodity WiFi devices," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 77–89.

[36] S. Tan and J. Yang, "WiFinger: Leveraging commodity WiFi for fine-grained finger gesture recognition," in *Proc. 17th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2016, pp. 201–210.

[37] E. TC-SMG, "Digital cellular telecommunications system (phase 2 +)," *General Packet Radio Service, Service description, Stage*, vol. 2, 1996.

[38] TensorFlow, "TensorFlow lite," 2018. [Online]. Available: https://www.tensorflow.org/lite/

[39] T. Tran, T. Pham, G. Carneiro, L. Palmer, and I. Reid, "A Bayesian data augmentation approach for learning deep models," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 2794–2803.

[40] Y.-C. Tung, D. Bui, and K. G. Shin, "Cross-platform support for rapid development of mobile acoustic sensing applications," in *Proc. 16th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2018, pp. 455–467.

[41] J. Wang, D. Vasisht, and D. Katabi, "RF-IDraw: Virtual touch screen in the air using RF signals," in *Proc. ACM Conf. SIGCOMM*, 2014, pp. 235–246.

[42] T. Wang, D. Zhang, Y. Zheng, T. Gu, X. Zhou, and B. Dorizzi, "C-FMCW based contactless respiration detection using acoustic signal," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 1, 2018, Art. no. 170.

[43] W. Wang, A. X. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 82–94.

[44] Y. Wang and Y. Zheng, "TagBreathe: Monitor breathing with commodity RFID systems," *IEEE Trans. Mobile Comput.*, vol. 19, no. 4, pp. 969–981, Apr. 2020.

[45] Y. Wang and Y. Zheng, "Modeling RFID signal reflection for contact-free activity recognition," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 2, no. 4, Dec. 2018, Art. no. 193. [Online]. Available: https://doi.org/10.1145/3287071

[46] M. Xu, M. Zhu, Y. Liu, F. X. Lin, and X. Liu, "DeepCache: Principled cache for mobile deep vision," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 129–144.

[47] J. Yang *et al.*, "Detecting driver phone use leveraging car speakers," in *Proc. 17th Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 97–108.

[48] K. Yang *et al.*, "cDeepArch: A compact deep neural network architecture for mobile sensing," in *Proc. 15th Annu. IEEE Int. Conf. Sens. Commun. Netw.*, 2018, pp. 1–9.

[49] S. Yun, Y.-C. Chen, H. Zheng, L. Qiu, and W. Mao, "Strata: Fine-grained acoustic-based device-free tracking," in *Proc. 15th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2017, pp. 15–28.

[50] X. Zeng, K. Cao, and M. Zhang, "MobileDeepPill: A small-footprint mobile deep learning system for recognizing unconstrained pill images," in *Proc. 15th Annu. Int. Conf. Mobile Syst. Appl. Serv.*, 2017, pp. 56–67.

[51] H. Zhang, W. Du, P. Zhou, M. Li, and P. Mohapatra, "DopEnc: Acoustic-based encounter profiling using smartphones," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 294–307.

[52] B. Zhou, J. Lohokare, R. Gao, and F. Ye, "EchoPrint: Two-factor authentication using acoustics and vision on smartphones," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 321–336.

[53] P. Zhou, Y. Zheng, and M. Li, "How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Serv.*, 2012, pp. 379–392.

Yanwen Wang (Member, IEEE) received the BS degree in electronic engineering from Hunan University, Changsha, China, the MS degree in electrical engineering from the Missouri University of Science and Technology, Rolla, Missouri, in 2010 and 2013, respectively, and the PhD degree from the Department of Computing, Hong Kong Polytechnic University, Hong Kong, in 2019. He is currently working as a postdoctoral fellow at the IMCL Lab, Hong Kong Polytechnic University, Hong Kong. His research interests include mobile and network computing, RFID systems and acoustic sensing.

Jiaxing Shen received the BE degree in software engineering from Jilin University, China, in 2014 and the PhD degree in computer science from The Hong Kong Polytechnic University, Hong Kong, in 2019. He is currently a research assistant professor at the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His research interests include mobile computing, data mining, social computing, affective computing, and Internet of Things. He has published several papers in high-impact journals and top conferences. He served as a reviewer for many international conferences and journals.

Yuanqing Zheng (Member, IEEE) received the BS degree in electrical engineering and the ME degree in communication and information system from Beijing Normal University, Beijing, China, in 2007 and 2010 respectively, and the PhD degree from the School of Computer Engineering, Nanyang Technological University, Singapore, in 2014. He is currently an associate professor at the Department of Computing, Hong Kong Polytechnic University, Hong Kong. His research interests include mobile and network computing, acoustic and wireless sensing, and Internet of Things (IoT). He is also a member of the ACM.