Making Sense of Doppler Effect for Multi-Modal Hand Motion Detection

Wenjie Ruan, *Member, IEEE*, Quan Z. Sheng, *Member, IEEE*, Peipei Xu, *Student Member, IEEE*, Lei Yang, *Member, IEEE*, Tao Gu, *Senior Member, IEEE*, and Longfei Shangguan, *Member, IEEE*

Abstract—Hand gesture is becoming an increasingly popular means of interacting with consumer electronic devices, such as mobile phones, tablets and laptops. In this paper, we present AudioGest, a device-free gesture recognition system that can accurately sense the hand in-air movement around user's devices. Compared to the state-of-the-art techniques, AudioGest is superior in using only one pair of built-in speaker and microphone, without any extra hardware or infrastructure support and with no training, to achieve multi-modal hand detection. Specifically, our system is not only able to accurately recognize various hand gestures, but also reliably estimate the hand in-air duration, average moving speed and waving range. We achieve this by transforming the device into an active sonar system that transmits inaudible audio signal and decodes the echoes of hand's movement at its microphone. We address various challenges including cleaning the noisy reflected sound signal, interpreting the echo spectrogram into hand gestures, decoding the Doppler frequency shifts into the hand waving speed and range, as well as being robust to the environmental motion and signal drifting. We extensively evaluate our system on three electronic devices under four real-world scenarios using overall 3,900 hand gestures collected by five users for more than two weeks. Our results show that AudioGest detects six hand gestures with an accuracy up to 96 percent. By distinguishing the gesture attributions, it can provide more fine-grained control commands for various applications.

Index Terms—Hand gesture recognition, device-free, audio signal, sonar, segmentation, FFT normalization

1 Introduction

The booming of consumer electronic devices has greatly stimulated the research on novel human-computer interactions. Hand gestures are a natural form of human communication with devices that have aroused enormous attentions from both industry and academia [1], [2]. Researchers and companies try to integrate the hand-gesture recognition into our daily devices, including laptops [3], tablets [4], and smartphones [5]. However, a crucial prerequisite of these applications is that the device can accurately and robustly detect gestures anytime (e.g., poor light condition at night), anywhere (e.g., in rural area without wireless connection) in a device-free manner (e.g., no need to wear extra devices/sensors) [4].

Over the last decade, many state-of-the-art hand gesture recognition (HGR) systems have been developed using various hardware platforms, such as computer vision [6], inertial sensors [7], ultrasonic sensors [3], infrared sensors (e.g., Leap Motion), and depth sensors [8]. While promising, most of these systems, however, can only partially meet those requirements [1]. For example, vision-based techniques are sensitive to the light conditions (i.e., performance greatly decreases in poor lighting conditions), and are usually regarded as privacy-intrusive. Although some commercialized HGR systems (such as Kinect, Leap Motion) achieve enormous success, their applications are still limited in computers and also need relatively high installation and instrumentation overhead (around 50∼250 USD). The wearable sensor based approaches (e.g., attaching 3-axis accelerometers or gyroscopes on hand) unavoidably require the user to wear additional devices. Although those systems can achieve fine-grained and multi-level hand motion detection in high precision, they may not be practical in real-world applications (e.g., user may feel uncomfortable or forget to wear the devices).

Many WiFi-based solutions have recently been proposed to overcome the above limitations. For example, WiGest [1] exploits the influence of in-air hand movement on the wireless signal strength of the device from an access point to recognize the performed gestures. Melgarejo et al. [9] leverage a directional antenna and WARP board to access various wireless features such as Received Signal Strength (RSS), signal phase differences and CSI (channel state information), then through matching the features from users' gestures with a standard set of pre-trained templates to recognize user's hand gestures. WiSee [10] exploits the doppler shift in narrow bands extracted from wide-band OFDM (orthogonal

- *W. Ruan is with Department of Computer Science, University of Oxford, Oxford OX1 3QD, United Kingdom. E-mail: wenjie.ruan@cs.ox.ac.uk.*
- *Q.Z. Sheng is with Department of Computing, Macquarie University, North Ryde, NSW 2109, Australia. E-mail: michael.sheng@mq.edu.au.*
- *P. Xu is with School of Electronic Engineering, UESTC, Chengdu 610051, China. E-mail: peipei.xu@adelaide.edu.au.*
- *L. Yang is with Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong. E-mail: young@tagsys.org.*
- *T. Gu is with School of Science, RMIT University, Melbourne, VIC 3000, Australia. E-mail: tao.gu@rmit.edu.au.*
- *L. Shangguan is with Computer Science Department, Princeton University, Princeton, NJ 08540. E-mail: longfeis@cs.princeton.edu.*

Manuscript received 31 Aug. 2016; revised 3 Sept. 2017; accepted 2 Oct. 2017. Date of publication 13 Oct. 2017; date of current version 2 Aug. 2018. (Corresponding author: Tao Gu.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2017.2762677

frequency-division multiplexing) transmissions to recognize nine different human gestures. Although WiFi-based systems can work under any lighting conditions and do not require dedicated hardware modification, those systems, however, require the mobile device to be always connected to a wireless transmitter/receiver, which is impractical for some circumstances such as on a train/bus or traveling in a rural area.

To tackle these challenges, we develop AudioGest, a device-free system that can transform consumer device into an active sonar system by utilizing the embedded microphone and speaker of the mobile device. Compared to other HGR systems, AudioGest exploits only one pair of built-in speaker and microphone without adding any extra cost on hardware. AudioGest does not require the model-training to achieve multi-modal hand gesture detection. The system not only can recognize hand gestures but also is able to accurately estimate the hand in-air time, average waving speed, and the hand moving range. We call such capability as *multi-modal* hand motion detection.

Implementing such a practical system, however, requires addressing a number of non-trivial challenges. First, the ambient noise (e.g., human conversation, electronic noise) dominates the recorded audio signals (see the experiments in Section 4.1). It is hence difficult to perceive the weak Doppler frequency shifts, let alone decoding the hand waving directions, speed, and range. Another challenge is the signal drifting brought by the device diversity and time elapse (see the experiments in Section 4.2). Since we emit a high-frequency audio signal ($> 18kHz$, making it inaudible to human), the Operational Amplifier (OA) in microphone and speaker both experience attenuation, making the magnitude of recorded echoes unstable. Moreover, different microphones/speakers have various OA attenuations, also resulting in signal drifting.

In AudioGest, we propose three main techniques to tackle the aforementioned challenges. First, we introduce an FFT-based normalization that substantially adjusts the magnitude of FFT frequency bin in different timestamps to the same level, removing the influence of OA attenuation in high-frequency part (see details in Section 6.1). We then perform *Squared Continuous Frame Subtraction*, in which we first subtract the spectrum of current audio frame by previous frame and square the magnitudes of frequency bins, further eliminating the nearby human motion influence (see details in Section 6.2.1). Furthermore, we apply a Gaussian smoothing filter [11] to transfer the discrete shifted frequency bins into a contouring area. We decode it into the real-time hand moving velocity curve based on the Doppler frequency shift (see details in Section 6.4). Finally, according to the velocity curve, we estimate hand gesture, moving speed, and waving range (see details in Section 6.5). In a nutshell, our main contributions are summarized as follows:

- We introduce an approach that utilizes one pair of COTS microphone and speaker to accurately detect the hand movement and to estimate fine-grained hand waving attributes. Our in-situ experiments with five users over a period of two weeks demonstrate the feasibility and accuracy of AudioGest in various living environments.
- We propose a denoising pipeline that not only abstracts the Doppler frequency shifts from weak echo signals, but also deals with the signal drifting issue caused by hardware diversity and time elapse.
- AudioGest is a training-free system that accurately recognizes 6 hand gestures with an accuracy of 95.1 percent on average, precisely distinguish the magnitude differences of various hand speed and moving range, providing up to 54 control commands by randomly choosing two attributes.

2 Related Work

Existing HGR systems can be categorized into two groups: *wearable sensor/device based* gesture recognition and *device-free* gesture recognition.

Wearable Devices based Gesture Recognition: Wearable sensor/device based systems utilize various sensors (i.e., 3-axis accelerometer [12], inertial sensor [13], or other smart devices [14]) to sense the movement of hands or arms. For example, some researchers infer the hand movement by wearing a shaped magnet [15]. Humantenna [13] requires users to wear a small Wireless Data Acquisition Unit (WDAU) enabling the human body as an antenna for sensing whole-body gestures.

Recently, Lu et al. [16] design a wearable device to acquire acceleration and SEMG (Surface ElectroMyoGraphic) signals and adopt a DTW-based Bayesian classifier to recognize 19 predefined gestures. More lately, some researchers adopt micro-radars to realize a series of gesture recognition applications. For instance, Li et al. propose Tongue-n-Cheek [17], a contact-less tongue gestures recognition system by designing a head-wearable device containing three 24G Hz micro-radars. All these gesture recognition systems either require users to wear a device/sensor (e.g., magnet ring, smart bracket and SEMG sensors) or need to install extra hardware such as WDAU, micro-radar or capacitive plates, which might add extra cost.

Beside those conventional gesture systems, some other research efforts focus on stroke-gesture recognition which enables smart-phones to accurately recognize the hand strokes on the screen. For example, Wobbrock et al. [18] develop a uni-stroke gestures recognition system, called $1 Recognizer, which can recognize 16 pen-gestures on the screen of a smartphone. Li et al. design Protractor [19], a fast and lightweight single-stroke gesture recognition system, which introduces a novel closed-form solution for calculating the similarity of hand strokes. However, these recognition systems are mainly for recognizing stroke-based gestures by touching the screen, which is different from our HGR system that focuses on in-air multi-modal hand gesture recognition without screen-touching.

Device-free Gesture Recognition: This category can be further classified into vision-based, environmental sensor based, RF-based, and sonar-based approaches. Video-based hand-gesture recognition systems often do the hand-region segmentation using color and/or depth information, and use the sequences of features for dynamic gestures to train classifiers, such as Hidden Markov Models (HMM) [20], conditional random fields [21], SVM [22], DNN [23]. However, vision-based techniques are usually privacy-invasive.

They also require users within the LOS (line of sight) of cameras, fail to work in dimmed environments, and incur high computational cost. Some environmental sensor-based hand recognition systems have emerged, such as Leap Motion that explores multiple channels of reflected infrared signals to identify hand gestures, and Kinect [24] that uses depth sensor to enable in-air 3D skeleton tracking.

Recently, RF-based gesture recognition systems are very popular due to their low-cost and being less intrusive [1]. WiVi [25], [26] uses ISAR technique to track the RF beam, enabling a through-wall gesture recognition. RF-Care [27] recognizes human gestures in a device-free manner based on a passive RFID (Radio-frequency identification) array. WiSee [10] can exploit the Doppler shift in narrow bands in wide-band OFDM (Orthogonal Frequency Division Multiplexing) transmissions to recognize 9 different human gestures. WiGest [1] explores the effect of the in-air hand motion on the RSSI in WiFi to infer the hand moving directions and speeds. Melgarejo et al. [9] leverage the directional antenna and short-range wireless propagation properties to recognize 25 standard American Sign Language gestures. AllSee [4] designs a very power-efficient hardware that extracts gesture information from existing wireless signals.

SonarGest [28] is one of the pioneering audio-based hand recognition systems, which uses three ultrasonic receivers and one transmitter to recognize 8 hand gestures. However, it needs to collect training data (potentially labour-intensive and time-consuming) and requires extra sonic hardware. SoundWave [3] is another HGR system exploiting audio Doppler effect. It only utilizes the built-in speakers and microphones in computers and requires no training. SoundWave designs a threshold-based dynamic peak tracking technique to effectively capture the Doppler shifts, and can distinguish five different hand gestures.

Most recently, researchers are trying to transform COTS speakers and microphones into a sonar system to detect human breath [29], to track a finger movement [30], and to sense user's presence [31]. Most of these systems adopt similar ideas from RF-based approaches, either decoding the echo of FMCW sound-wave to measure the human body, or utilizing the OFDM to achieve real-time finger tracking, or exploring the Doppler effect when human approaching or away from the microphone. However, such systems need two microphones or require specialized design of sound-wave that is power-intensive. Different to previous works, our work only uses one speaker and one microphone by emitting single-tone audio to achieve multi-modal gesture recognition.

3 Preliminaries

3.1 Doppler Effect

Most of current HGR systems utilize labeled sensor readings (including images) to train a classification model, and then distinguish hand gestures, which is lack of physical interpretation. It is also hard for those systems to detect some context information regarding the hand gestures, such as hand's moving speed and in-air waving duration. AudioGest system in this paper, conversely, is inspired by a prevalent law in the physical world namely the *Doppler Effect*.

Doppler effect illustrates and quantifies the wavelength changes when wave energy like sound or radio waves travel between two objects if one or both of them move. The Doppler effect causes the received frequency of a source to differ from the sent frequency if there is motion that is increasing or decreasing the distance between the source and the receiver. The general equation of measuring frequency shift is as follows:

$$\Delta f = \frac{\Delta v}{v_{wave}} f_{source}, \tag{1}$$

where $\Delta f = f_{receiver} - f_{source}$, called *Doppler Frequency Shift*; $\Delta v = v_{receiver} - v_{source}$, is the velocity of the receiver relative to the source: it is positive when the source and the receiver are moving towards each other.

In our case, the wave source (i.e., speaker) and the receiver (i.e., microphone) are both motionless but the reflector (i.e., human hand) are moving. Hence, though most of sound waves stay unchanged, a part of acoustic waves that is reflected by a moving hand experiences a Doppler frequency shift measured by Eqn. (2)

$$f_{received} = \frac{1 + v_{rad}/v_{sound}}{1 - v_{rad}/v_{sound}} f_{sound}, \tag{2}$$

where v_{rad} means the radial speed of hand to microphone. Such Doppler effect caused by the motion of a reflector is widely adopted in modern radar systems or underwater sonar systems. Motivated by this intuition, *AudioGest* aims to sense such doppler frequency shift of weak reflected acoustic waves by a moving hand. As shown in Fig. 1, when a hand moves in different directions or at different speeds, it will cause different Doppler frequency shifts (e.g., different shapes, different intensities and durations). Our AudioGest targets to decode such Doppler frequency shifts, to recognize the gestures, and to estimate the moving speed and duration of a hand in air.

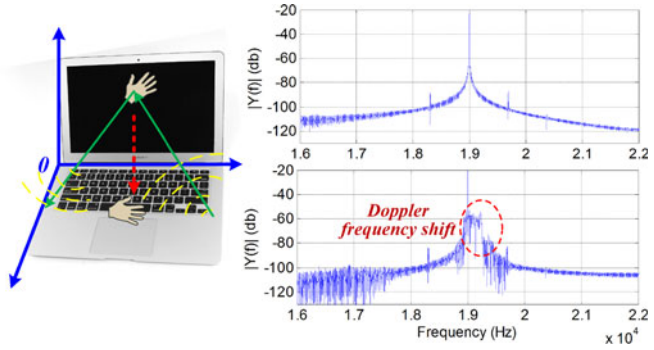


Fig. 1. Illustration of doppler frequency shift.

3.2 COTS Speakers & Microphones

In this paper, we aim to turn the COTS speakers and microphones into an active sonar system to detect fine-grained hand gestures without annoying normal human audition.

Normally, human audible signal lies between 20 Hz∼18 kHz. Assuming that maximum hand waving speed is less than 4 m/s, it requires 0.47 kHz extra bandwidth (under a sampling rate of 44.1 kHz, see Section 6 for details on how to calculate the frequency bandwidths). As a result, the speakers and microphones needed should be at least with a capability of up to 18.47 kHz frequency-response.

Fig. 2. Speakers and microphones in COTS mobile devices.

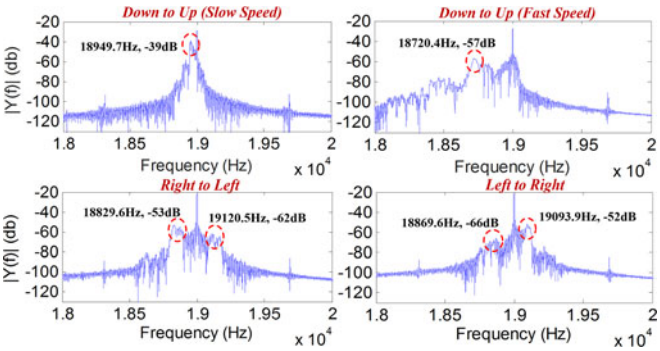


Fig. 3. The Doppler frequency shifts caused by different hand gestures and waving speeds.

According to the NyquistShannon sampling theorem, to accurately recover a 20 kHz signal, the microphones at least support a 40 kHz sampling rate. Fortunately, mobile devices are increasingly supporting high-definition audio capabilities targeted at audiophiles. In particular, such advancement includes high-frequency response range, microphone arrays for stereo recording and noise cancellation, and $4\times$ improvement in audio sampling rates. Fig. 2 shows COTS microphones and speakers of three typical mobile devices. They all can support up to 22 kHz response frequency and typical 44.1 kHz or 48 kHz sampling rate, making it possible to achieve fine-grained hand detection.

4 Empirical Studies and Challenges

4.1 Weak Echo Signal

As Fig. 1 shows, we transmit a 19 kHz sine acoustic wave (for 3 s) from the right channel of the speaker in a laptop (i.e., MacBook Air). Simultaneously, we record the ambient sound signal using a microphone. At the same time, a participant waves his hand in different directions and speeds. Then we conduct an FFT to see the frequency shift of audio signal caused by hand motion.

From Fig. 3, we can observe that the waving hand from down to up results in an observable magnitude increase in the lower frequency bins, but moving hand from left-to-right/right-to-left is less obvious and the echo signal is weak (i.e., the bins marked by the red circles, the left sides of 19 kHz bin). Specifically, we find that the motion speed of the hand is highly related with the location of such increased frequency bins, i.e., moving hand in a slow speed causes a risen magnitude in 18,949.7 Hz bin, but with a fast speed, it leads to an increase in 18,720.4 Hz bin. Also, moving hand from right to left and left to right will arouse a frequency shift in both sides but with opposite intensities (e.g., -53 dB and -62 dB for right-to-left, -66 dB and -52 dB for left-to-right).

In summary, such observable frequency shifts highly motivate our AudioGest system but also bring us a challenging task—how to abstract such weak, vulnerable frequency-bin changes from wideband[1] audio signals. Moreover, we intend to decode the fine-grained hand moving speed, in-air duration and motion range beside the hand-gesture recognition. With ambient noise (such as human conversation, electronic noise and environmental sound), it is even harder for us to perceive these Doppler frequency shifts. We will illustrate our solution in Section 6.2.

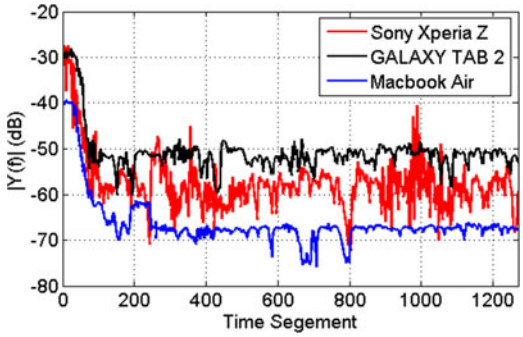


Fig. 4. The sound signal drifts for different mobile devices at different time slots.

4.2 Audio Signal Drift

Another challenge is about the audio signal drifts, which can be categorized into two types: *i)* temporal signal drift: audio signals received in different time slots depict various magnitudes for a same frequency bin; and *ii)* diverse-device signal drift: audio signals record by different microphones reveal various magnitudes for the same frequency bin.

Fig. 4 illustrates the experiment we conducted under a *static* environment,[2] where microphones from various types of mobile devices record 1-hour reflected audio signals while speakers of the same device continuously emit 19 kHz inaudible sinuous sound-waves. We divide the 1-hour soundwave into 1,270 signal frames, and further apply 2,048-point FFT. We plot the strengths of frequency bin at 19 Hz over the time for three different mobile devices in Fig. 4. We find that for different mobile devices, the frequency magnitudes are diverse. Even for a same electronic device, the signal strengths fluctuate over the time, and the mobile phone exhibits a stronger signal drift. We also observe that the recorded audio signals drop significantly during first 10 minutes, which lies on two reasons. One reason is that the OA is the main component of the speaker and microphone, and emitting high-frequency sound-waves (i.e., 19 kHZ audio signal) will let the OA work on the upper-boundary of its capability, thus is unstable. The other reason is that with the time evolving, continuous ringing of the speaker generates a fair amount of heat that increases the working temperature of the electronic components, especially in the first 10 minutes when speakers just start to work. It is well known that the electronic device is very sensitive to temperature, which influences the performance of the speaker. Such signal drifting will greatly hinder the system's scalability, which means an HGR approach

1. Normally, a microphone can resolve 0~22.05 kHz sound signal for a 44.1 kHz sampling rate.

2. *Static* means no hand moving, same meaning applied in the rest of the paper.

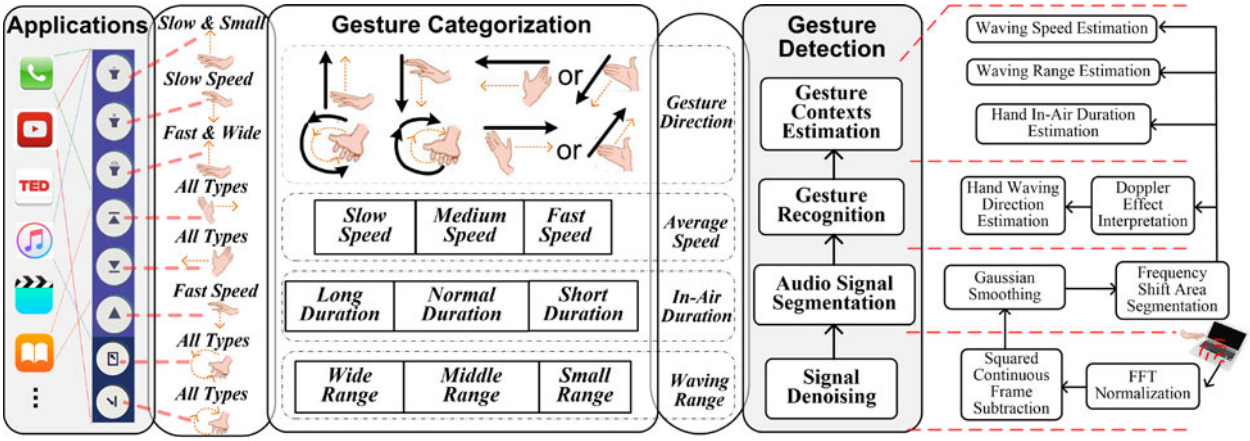


Fig. 5. Overview of the system for hand gesture detection.

that works well in one device may be incapable for other devices or in different time-slots. We will deal with this challenge in Section 6.1.

5 System Conceptual Overview

This section will introduce the system architecture of AudioGest, mainly including three conceptual layers—the *gesture detection* layer, the *gesture categorization* layer, and the *application* layer, as shown in Fig. 5.

The gesture detection layer is the key part of the whole system (the details shown in the right part of Fig. 5). This layer outputs four kinds of gesture contexts—*waving direction, hand's average speed and in-air duration*, as well as *waving range*. Specially, to detect such fine-grained gesture features, we first eliminate the noise of received raw acoustic signal which contains two steps - FFT normalization and background noise subtraction (i.e., dealing with the *Audio Signal Drift* challenge). Then, we need to accurately identify the audio signal segments caused by hand's motion, consisting of two parts—Gaussian smoothing and segmenting the frequency shift area (i.e., tackling the *Weak Audio Signal* challenge). In the next, based on the magnitude changes and temporal locations of segmented frequency bins, we interpret such Doppler frequency shifts, thus estimate the hand waving directions. Finally, we put things together, further quantify the hand in-air durations, waving ranges and average speeds.

The gesture categorization layer categorizes different basic gesture characteristics from previous layer into different semantics. As Fig. 5 shows, we define overall six gesture directions and three intensity levels for the moving speed, in-air duration and waving range. Unlike previous systems that only detect one or two hand gesture contexts [1], [4], AudioGest provides three types of hand motion attributes except the basic hand gestures. By randomly choosing two motion attributes, AudioGest can theoretically provide up to $6 \times 3 \times 3 = 54$ control commands, which we thus call *multi-modal* hand gesture recognition. It is noted that AudioGest can support a more fine-grained categorization (e.g., classify the in-air duration into four or five levels) which leads to more control commands but degrade the detection accuracy possibly. Vice-versa, we can use a course-grained categorization to increase the estimation accuracy. For example, for an e-book application (only needs 4 commands, *next page*, *previous page*, *full screen*, *normal screen*), we can choose four types of hand waving directions (regardless of waving speed, in-air duration and range) to control these command buttons. This layer provides flexible controlling choices to the application layer.

The application layer maps different gestures to control commands for various applications. Typically, one action is mapped to one gesture type and the developer can pick one or more hand gestures to represent an action. For example, for a media player application, a *play* action can be performed with a *Up-Down* hand gesture while a *volume up* action can be mapped to moving the hand up. The volume changing rate can be controlled by the speed or range of the hand waving.

6 Realizing the AudioGest System

In this section, we will illustrate how to achieve gesture detection and address the associated challenges. Before that, we first introduce how to design the transmitted audio signal. Human normal audible scope is 20 Hz$\sim$18 kHz. To avoid annoying human audibility, under no circumstance, should AudioGest produce the sound signal below 18 kHz (to be more safe, we make it 18.5 kHz). Assuming that the fastest hand moving speed is 4 m/s [3], then the largest Doppler frequency shift $\Delta f_{doppler} = (2v_{hand}/v_{sound})f_{transmit} = 470.6$ Hz. Hence, if the mobile device transmits a 19 kHz sound, then the received audio signal is 18,529.4 Hz$\sim$19,470.6 Hz, satisfying the requirement.

6.1 FFT Normalization

Since our targeted sound frequency band is 18.5 kHz$\sim$19.5 kHz, intuitively, we may need a band-pass filter or high-pass filter. However, the introduced FFT normalization is based upon the frequency domain of the recorded audio signal. We only perform analysis to the FFT bins within the targeted narrow bandwidth. Such processing will naturally filter out the influence of audible noise without adding an extra signal filter.

In order to observe how the Doppler frequency shifts along the time, we first adopt a 2,048-point hamming window to segment the filtered signal into audio frames,[3] then

3. Each frame represents $2{,}048/44{,}100 = 0.0464$ s audio signal.

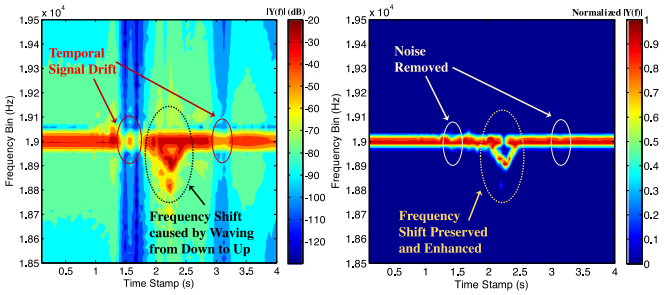


Fig. 6. Left: Raw audio spectrogram; Right: Audio spectrogram after FFT normalization.

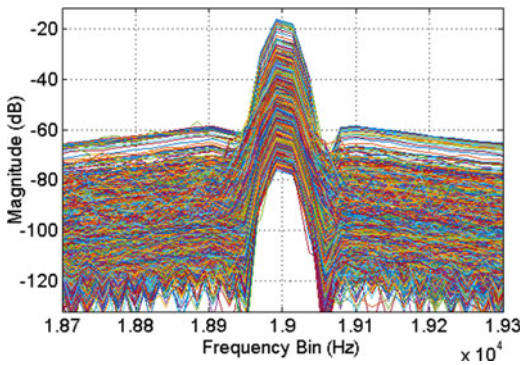


Fig. 7. All spectrums of audio signal frames: Each line represents a spectrum of each frame.

apply a 2,048-point FFT[4] to each frame to get the sound spectrogram, shown as the left graph in Fig. 6. We can see the signal drift severely interferes the audio spectrogram, displaying an unstable magnitude (e.g., the part marked by the red ellipses).

To deal with this challenge, we collect 3,600 seconds 19 kHz sound signal using three different mobile devices and then segment the signal into frames of 2,048-point length. As Fig. 7 shows, we plot the spectrum of 78,260 audio frames in the same graph. We can observe that, although the magnitude of the frequency bins for different frames show unpredictable signal excursions (e.g., the magnitude in 19 kHz bin spans from -83 dB~ -24 dB), the relative magnitudes for every single sound frame are stable and robust to the time-elapse and device diversity (i.e., each spectrum shows a similar shape). Because we intend to perceive the Doppler frequency shifts to infer hand gestures, we are more concerned about how the peak frequency bin changes over the time instead of absolute magnitude of each frequency bin. Based on this intuition, we normalize the magnitudes of frequency bins for each audio frame. Shown in the right graph of Fig. 6, after a simple FFT-based normalization, the audio spectrograms produced by waving hand from *Down to Up* show a stable and interpretable Doppler frequency shift and the signal drift is removed.

6.2 Audio Signal Segmentation

6.2.1 Squared Continuous Frame Subtraction

To perceive the magnitude changes of frequency bins, we further conduct a *Squared Continuous Frame Subtraction*, in

4. With a 44.1 kHz sampling rate, the velocity detection resolution $v_{res} = (f_s/FFT_{points})(v_{sound}/f_{source} = 0.39$ m/s.

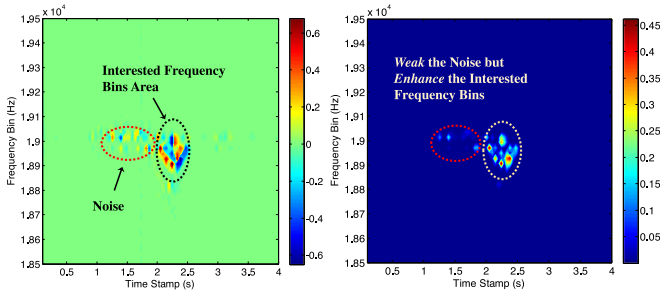


Fig. 8. Left: The spectrogram after continuous frame subtraction; Right: The spectrogram after the square calculation.

which we first subtract the normalized spectrum of current audio frame by previous frames and then square the magnitudes of frequency bins. The continuous subtraction essentially eliminates the static frequency bins and save the changed bins, shown as the left graph in Fig. 8 (i.e., remove the unchanged 19 kHz bin in Fig. 6 and highlight the changed frequency bins). The square calculation will further enhance the frequency-bin changes caused by hand's movement but weak the bins due to the noise (see the right graph in Fig. 8, the noise marked by the red dot oval is further eliminated). In the next, we need to accurately segment the frequency shift area based on those discrete frequency bins.

6.2.2 Gaussian Smoothing

Revisit the right graph of Fig. 8, the x-axis represents the time-stamps in a 0.046 second resolution, the y-axis indicates the frequency bins in Hz, the colors ranging from blue to red quantify the changing magnitude of frequency bins. Intuitively, we thereby can view such spectrogram graph as an image, then what we are interested is to connect those pixels and augment it into a zone. To do so, we introduce a Gaussian Smoothing method to blur the whole image. The Gaussian smoothing is a type of image-blurring filter that uses a Gaussian function for calculating the transformation to apply to each pixel in an image. Specifically, each pixel's new value is set to a weighted average of that pixel's neighborhood. The original pixel's value receives the heaviest weight (having the highest Gaussian value) and neighboring pixels receive smaller weights as their distance to the original pixel increases. For our two-dimensional image, the following function is used for smoothing

$$G(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right), \tag{3}$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. Intuitively, this formula produces a surface whose contours are concentric circles with a Gaussian distribution from the center point, which preserves boundaries and edges well. As the left graph in Fig. 9 shows, after Gaussian smoothing, those peak pixels are well augmented into a zone. Furthermore, we set a threshold ω to conduct the image binarization, i.e., set the pixel value to zero if its value is less than ω, set the pixel value to one otherwise.

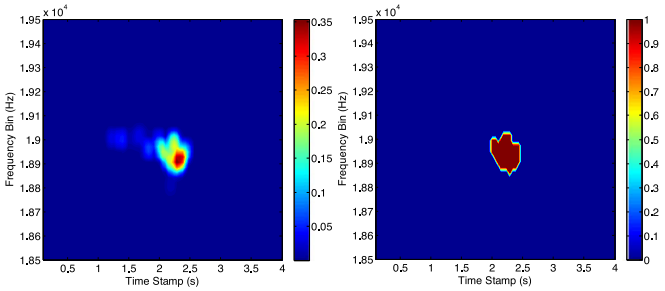


Fig. 9. Left: The spectrogram after Gaussian Smooth filter; right: The segmented area where Doppler frequency shift happens.

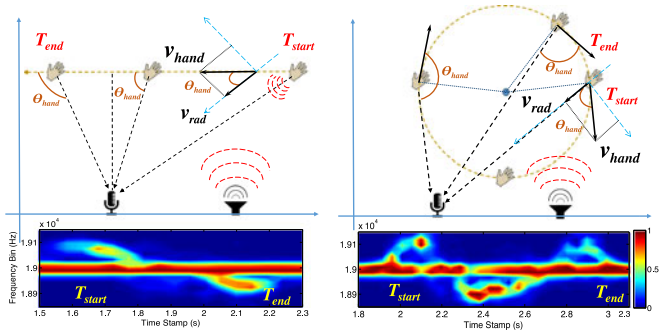


Fig. 10. The hand moving path with its generated audio spectrogram. Left: Hand moving from Right to Left; Right: Hand moving along clockwise circle.

As shown in the right graph of Fig. 9, we can successfully segment the frequency zone that Doppler shift happens. More de-noising and segmentation examples can be found in Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TMC.2017.2762677.

6.3 Doppler Effect Interpretation

In this section, by using two typical hand-waving examples, we will interpret how a hand movement generates the shifted audio spectrogram based on the *motion law* of the hand movement.

From Eqn. (2), since $v_{sound} \gg v_{rad}$, we have

$$\Delta f = \frac{2 f_{sound} v_{rad}}{v_{sound}}, \tag{4}$$

where $\Delta f = f_{received} - f_{sound}$. As Fig. 10 shows, assuming that hand moving path has θ_{hand} with the microphone and the hand moving speed is v_{hand}, we have

$$v_{rad} = v_{hand} \cos\theta_{hand}. \tag{5}$$

Furthermore, we can derive the relation based on Eqns. (4) and (5) as follows:

$$\Delta f = \frac{2 f_{sound} v_{hand} \cos\theta_{hand}}{v_{sound}} \propto v_{hand} \cos\theta_{hand}. \tag{6}$$

We take two examples to interpret Eqn. (6), showing how we link real-time hand moving gesture with the audio spectrogram. As Fig. 10 depicts, when the hand moves from *Right to Left*, θ_{hand} gradually increases (e.g., from $\pi/6$ to $\pi/2$ then to $2\pi/3$), hence the $\cos\theta_{hand}$ decreases[5] to 0, then to a negative value (e.g., from $\sqrt{3}/2$ to 0, then to $-1/2$). As a result, the frequency shifts from high-frequency (i.e., higher than 19 kHz) to zero, then to low-frequency (i.e., lower than 19 kHz). For the most complicated case *clockwise circle*, the θ_{hand} first decreases from a certain angle to zero, then gradually increases from zero to π, and then decreases from π to the previous angle (e.g., θ_{hand} experiences $\pi/3 \rightarrow 0 \rightarrow \pi/2 \rightarrow \pi \rightarrow \pi/3$ the right graph of Fig. 10). Thus, the audio frequency shifts towards high-frequency at first, then goes back to 19 kHz, further moves to the low-frequency, then it goes back to zero, continuously moves to high-frequency.

5. $\cos\theta$ is a monotony decrease function in $[0, \pi]$.

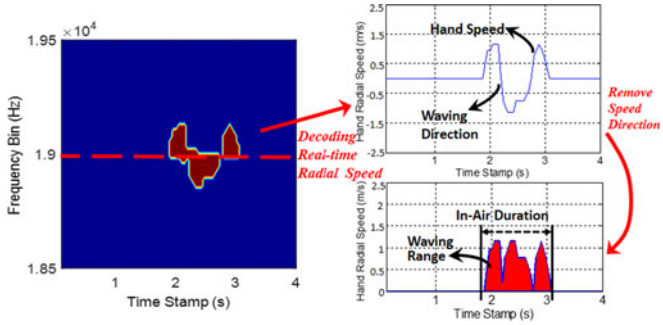


Fig. 11. The illustration of transforming frequency shifts into hand velocity, in-air duration and waving range.

6.4 Transforming Frequency Shift Area into Hand Velocity

This section will introduce how to estimate the real-time hand radial velocity based on the segmented frequency shift area. It should be noted that the peak bin locates in 19 kHz under a no hand-waving environment (using $v_0 = 0$ represents such case). Based on Eqn. (6), we can model the frequency shift with real-time hand radial velocity as

$$\begin{aligned} f_{received}(t) - f_{sound} &= \frac{2 f_{sound}}{v_{sound}} v_{hand}(t) \cos\theta_{hand}(t) \\ &= \frac{2}{\lambda_{sound}} v_{rad}(t). \end{aligned} \tag{7}$$

Furthermore, we can derive hand radial velocity $v_{rad}(t) = 0.5\lambda_{sound}(f_{shift}(t) - f_{sound})$. As the left graph of Fig. 11 shows, at each time-stamp, the length of frequency interval marked by red color represents $(f_{shift} - f_{sound})$. Therefore, we can estimate the real-time radial velocity of hand as shown in the right top graph in Fig. 11. Essentially, the sign of hand radial velocity indicates the hand moving direction (i.e., hand gesture type), and the time interval of non-zero velocity represents the hand in-air duration. Also, we can measure the hand waving range based on the area covered by the velocity curve.

6.5 Gesture Recognition

6.5.1 *Recognizing the Waving Direction*

Similarly, based on the direction changes of radial velocity (i.e., whether its value is negative or positive, determined by $\cos\theta_{hand}$), we hence can estimate the angle ranges of the hand movement (i.e., in angle categories: $[0, \pi/2]$ or $[\pi/2, \pi]$), as well as its corresponding time duration in each

TABLE 1
Rules of Recognizing Hand Waving Directions

Hand Waving Direction	Rules of Hand Motion Angle $\theta_{hand}(t)$	Rules of Motion Duration
Up to Down	$[0, \pi/2]$	N/A
Down to Up	$[\pi/2, \pi]$	N/A
Right to Left	$[0, \pi/2] \rightarrow [\pi/2, \pi]$	$t_{[0,\pi/2]} > t_{[\pi/2,\pi]}$
Left to Right	$[0, \pi/2] \rightarrow [\pi/2, \pi]$	$t_{[0,\pi/2]} < t_{[\pi/2,\pi]}$
Anticlockwise Circle	$[\pi/2, \pi] \rightarrow [0, \pi/2] \rightarrow [\pi/2, \pi]$	N/A
Clockwise Circle	$[0, \pi/2] \rightarrow [\pi/2, \pi] \rightarrow [0, \pi/2]$	N/A

angle category. Based on a sequence of angle categories and its durations, we can further detect different gesture types. AudioGest adopts a rule-based method to infer the types of hand gestures. These rules are originated from the interpretation of *Doppler Effect*, which first exploit the frequency shifting direction to decode $\cos\theta_{hand}$, then to further estimate θ_{hand}, i.e., the hand waving direction towards the microphone. Finally, based on the hand waving direction sequence $\theta_{hand}(t)$, we estimate the hand waving directions. We summarize the gesture recognition rules as shown in Table 1. The examples can be found in Appendix-B, available in the online supplemental material. It is noted that many hand-gestures recognition systems highly depend on semi-supervised/supervised machine learning methods [4]. Our AudioGest system does not need to collect labeled training data to train a classifier.

6.5.2 Estimating Waving Duration and Speed

For estimating the hand in-air duration, we can directly measure the time interval that hand radial velocity is not equal to zero (e.g., the time length marked by dot-line in Fig. 11). Then the remaining problem is how we measure the average hand moving speed. Please note that the velocity curve we estimate is the hand radial speed (towards the microphone). In this paper, we aim to first recognize different hand gestures, then to be able to distinguish different hand speed, in-air duration and moving range to provide more control commands for serving various applications. Hence, for a same gesture type, we want to evaluate if the hand is in slow, medium or fast speed (see Fig. 5).

Specifically, we first transfer the hand velocity (with moving direction) into a speed (ignore the direction), the transformation shows as the right-top graph to the right-bottom graph in Fig. 11. We observe that, for the same gesture with different speeds, θ_{hand} actually experiences a same angle range (e.g., $\pi/6 \rightarrow \ldots \rightarrow \pi/2 \rightarrow \ldots \rightarrow 2\pi/3$: moving from right to left as in the left graph of Fig. 10) but in different timestamps. As a result, according to Eqn. (5), we can infer that $E(V^1_{hand}) > E(V^2_{hand}) \Leftrightarrow E(V^1_{rad}) > E(V^2_{rad})$, where $V^1_{rad} = \{v^1_{rad}(t_1), v^1_{rad}(t_2), \ldots\}$ represents the first sequence of hand radial speed we estimated, V^2_{rad} indicates the second sequence of hand radial speed.[6] Hence we define a *speed-ratio* to evaluate the relative magnitude for different hand speeds. Assuming that the time interval between two adjacent timestamps is T (e.g., 0.0464 s using a 2,048-point frame), the hand waving duration is $t_{waving} = nT$, then we calculate the *speed-ratio* as

$$S_{ratio} = \frac{E(v_{rad}(t))}{E(v^0_{rad}(t))} = \frac{\frac{1}{n}\sum_{i=1}^{n} v_{rad}(iT)}{E(v^0_{rad}(t))}, \tag{8}$$

where $E(*)$ means expectation or mean value; $v^0_{rad}(t)$ represents a baseline of the hand moving speed set as $E(v^0_{rad}(t)) = 1$ for simplicity. Hence, we have $S_{ratio} = \frac{1}{n}\sum_{i=1}^{n} v_{rad}(iT)$, namely the mean value of our estimated radial-speed. Intuitively, a bigger S_{ratio} represents a faster hand movement.

6.5.3 Estimating Waving Range

By inheriting the idea in evaluating the waving speed, we also define *range-ratio* to measure the relative magnitude of hand waving range

$$R_{ratio} = \frac{R_{rad}}{R^0_{rad}} = \frac{\sum_{i=1}^{n} Tv_{rad}(iT)}{R^0_{rad}} = \frac{nTS_{ratio}}{R^0_{rad}}, \tag{9}$$

where R^0_{rad} represents the baseline of hand waving range that we assume equals to 1. Hence we can compare the hand waving ranges using $R_{ratio} = nTS_{ratio}$ (i.e., the area of the zone covered by red color in Fig. 11), where n and S_{ratio} is the estimated hand in-air duration and speed-ratio.

7 Evaluation

We start with micro-benchmark experiments in a lab environment and then conduct the in-situ tests in four real-world places—Living Room, Bus, Cafe, and HDR Office. We conduct the testing on three typical mobile devices: laptop (MacBook Air laptop), tablet (GALAXY Tab-2 tablet), and mobile phone (GALAXY S4 smartphone) without any hardware modification. We name the three devices as *D1*, *D2* and *D3* for simplicity.

Hardware. For the MacBook Air laptop, we run AudioGest on the computer using Audio System Toolbox.[7] For the GALAXY tablet and smartphone, we design the AudioGest system in the Simulink8.6 that provides a library of Simulink blocks for accessing the devices speaker and microphone.[8]

Testing Participants. Five participants join the experiments. AudioGest decodes the hand gesture via analyzing the reflected audio signal from hands. Intuitively, a bigger hand generates a stronger echo signal. Thus we measure the handsize of each participant (see Appendix D, available in the online supplemental material). The five users are marked as *U1*, *U2*, *U3*, *U4* and *U5*.

Evaluation Metrics. We adopt four typical evaluation metrics to evaluate our methods: *i)* Detection Rate (or True Detection Rate): the ratio of correctly detected hand gesture to overall testing hand gestures, measuring whether our system can efficiently detect a hand gesture when a hand waving happens; *ii)* False Detection Rate: the ratio of wrongly detected hand gestures to overall detected hand gestures, evaluating whether our system is too "sensitive" by

6. Essentially, V^1_{rad} and V^2_{rad} represent two different moving speeds for a same certain hand-gesture type.

7. mathworks.com/hardware-support/audio-ast.html

8. mathworks.com/hardware-support/android-programming-simulink.html

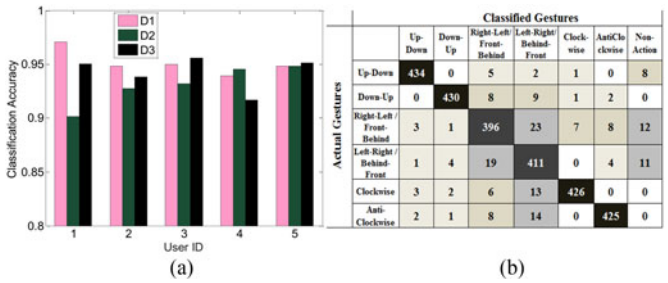


Fig. 12. (a) The average gesture classification accuracy for different mobile devices and users. (b) The confusion matrix for the gesture classification.

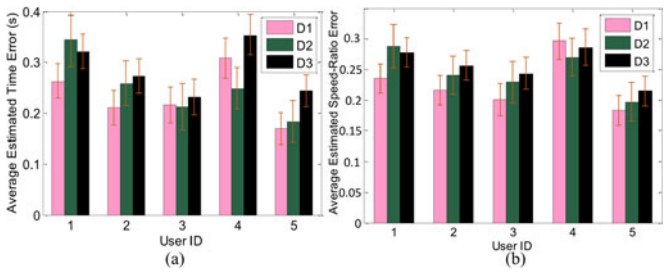


Fig. 13. (a) The hand in-air duration estimation error for different mobile devices and users. (b) The average speed-ratio estimation error of hand moving for mobile devices and users.

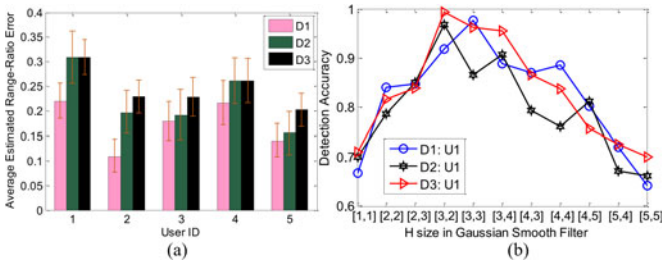


Fig. 14. (a) The average range-ratio estimation error of hand moving for different users. (b) The gesture detection accuracy with parameter H-size.

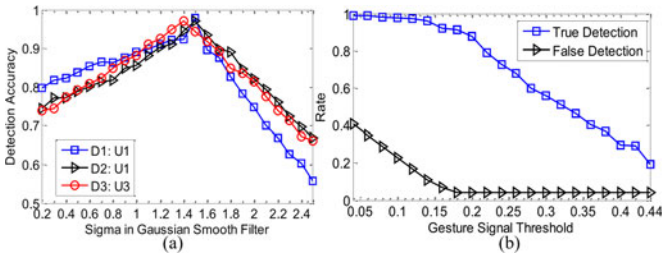


Fig. 15. (a) The gesture detection accuracy with parameter σ. (b) The gesture detection accuracy with gesture signal threshold.

recognizing a non-handgesture as a hand gesture; *iii)* Gesture Classification Accuracy: the rate that system can correctly classify the gesture type among all the detected hand gestures; *iv)* Detection Accuracy: the rate that system can correctly classify the gesture types as well as the categories of the in-air duration, average speed and waving range. We collect the ground truth by using a smart-watch with a 3-axis MEMS accelerometer (the details can be found in Appendix E, available in the online supplemental material).

7.1 Micro-Test Benchmark

We conduct some micro-benchmarks in a lab environment. We ask the five participants to perform each hand gesture 30 times for each device, hence we test 2,700 hand gestures by collecting around 4.5 hours audio data.

Gesture Recognition. Fig. 12 shows the gesture classification accuracies of five users for three devices. AudioGest achieves 94.15 percent gesture type recognition accuracy. In particular, subject U5 can get average 95 percent accuracy, but U1 achieves 90.15 percent mean accuracy using the tablet. From its confusion matrix (shown in Fig. 12), we can observe that most errors happen in distinguishing *Right-Left/Front-Behind* and *Left-Right/Behind-Front*. Detecting the hand gestures is done by decoding the hand-microphone angle sequence and its corresponding duration. For device D1 (i.e., MacBook Air laptop), its microphone locates in the left side, which results in different duration time of two angle categories for *Right-Left* and Left-Right waving. But we cannot distinguish hand waving from *Front-Behind* or *Behind-Front* due to the block of the computer screen. However, for D2 and D3 (i.e., Galaxy tablet and smartphone), their microphones locate in the bottom of the device, which substantially enable *Right-Left* and Left-Right hand movement generating the same angle category sequence (i.e., $[0, \pi/2] \rightarrow [\pi/2, \pi]$) and roughly same durations. Hence we cannot distinguish such two directions, but we can recognize the *Front-Behind* or *Behind-Front*. Due to the same reason, for recognizing *Right-Left/Front-Behind* and *Left-Right/Behind-Front*, we can only depend on the difference of angle durations, making it less reliable as other directions. Moreover, to better illustrate the idea of multi-modal hand detection, we depict several real-world examples in Appendix C, available in the online supplemental material.

Waving Attributes Estimation. Figs. 13 and 14 show the results of estimation errors of the hand in-air duration, moving speed-ratio and range-ratio respectively. The bar charts indicate both average error and its standard derivation. Specifically, AudioGest can estimate the three gesture context information with average 0.255 s in-air duration, 0.242 speed-ratio and 0.2138 range-ratio error respectively. It is worth to mention that, among 5 subjects, U5 achieves a better result in both gesture classification and context estimation, which mainly lie in the fact that U5 has a slightly bigger hand, which enhances the audio signal reflection.

Parameters Chosen. Figs. 14 and 15 illustrate how three key parameters influence the performance of our system. The parameter *H-size* specifies the number of rows and columns used in the gaussian filter (i.e., $H_{size} = [x, y]$ in Eqn. (3)). We test overall 11 different H-size when $[x = 3, y = 2]$ performs better. Parameter σ indicates the standard deviation in Gaussian function, which achieves the best accuracy at $\sigma = 1.5$. The last parameter *Gesture-Signal Threshold* determines whether a shift happens in a frequency bin, which plays an important role in AudioGest. We can see that the higher the value is, the more true detection and false detection rates decrease. Hence we choose *Threshold = 0.16* to balance such two detection rates.

As Table 2 shows, we also measure the FFT resolution, calculation time, speed, and in-air time detection resolutions by using different signal frame sizes. We find that for a smaller frame size, we need to calculate more FFTs within a second and get a smaller frequency bin, which in turn

TABLE 2
Calculation Time and Resolution versus Frame Sizes

Frame Size	Resolution of FFT (Hz)	Calculation Time (s)	Resolution of Speed (m/s)	In-Air Time Resolution (s)
256	172.27	2.767	3.110	0.0058
1,024	43.07	0.733	0.777	0.0232
2,048	21.53	0.396	0.389	0.0464
4,096	10.77	0.226	0.194	0.0929
8,192	5.38	0.134	0.097	0.1858

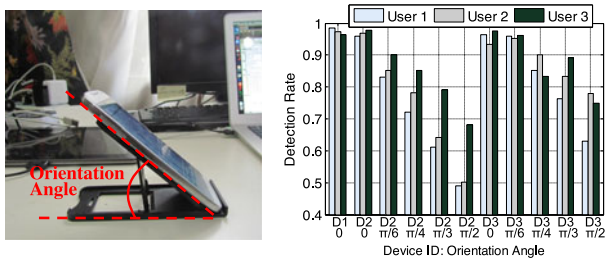


Fig. 16. The device orientation angle with its detection accuracy.

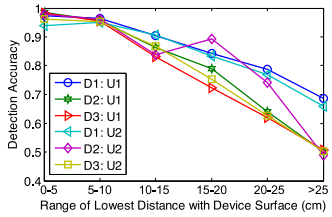


Fig. 17. The device-hand distance with its detection accuracy.

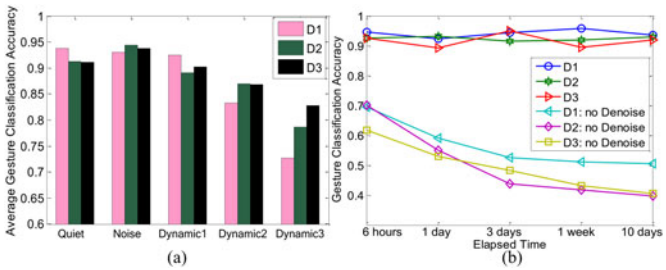


Fig. 18. (a) The average detection accuracy for different scenarios. (b) The detection accracy with and without denoising.

produces a finer speed resolution but a coarser time resolution. To balance the speed and time resolution as well as to maintain a reasonable calculation time, we choose 2,048 as the frame size and as the FFT points. Note that the speed resolution is also equivalent to the lower-boundary of hand speed that we can detect (e.g., if the hand speed is extremely slow such as less than 0.389 m/s, our HGR system cannot detect it). However, our system focuses on multi-modal hand gesture recognition, in which we categorize the hand speed into three levels: slow, medium, and fast (see Fig. 5). A speed resolution of 0.389 m/s is accurate enough to serve the purpose of this system because this resolution has a good trade-off among the calculation time, speed resolution and time resolution, especially it can filter out some false alarms caused by finger movements (those movements usually produce gentle frequency shifts which can be captured by a sensitive speed resolution).

Please note that we can also use a 4,096-point frame size that can reach 0.194 m/s speed resolution for a more fine-grained hand gesture detection (e.g., we can categorize the hand-speed into 4 or more ranges so that HGR system can provide more control commands). The choice of frame size mainly depends on the real-world applications (e.g., whether it requires a smaller delay, more fine-grained speed and in-air time detection) and the calculation capacities. The decision also relates to the sampling rate that a mobile device can support. For example, if the hardware supports a higher sampling rate (e.g., 192 kHz in SUMSUANG Galaxy S6 smart-phone), we can choose 1,024-point or even 512-point frame size to achieve a better or comparable speed resolution as 2,048-point size in 44.1 kHz sampling rate but with a better time resolution. In AudioGest, for generality, we set its sampling rate as 44.1 kHz. With this sampling rate, we choose 2,048-point frame size, which is acceptable for multi-module hand-gesture detection.

System Robustness. We evaluate the robustness of AudioGest in four ways:

- *Orientation Angle*: as Fig. 16 shows, AudioGest performs well when the orientation angle is less than $\pi/4$. Under a $\pi/2$ circumstance, its accuracy greatly decreases to around 60 percent, which we will leave for further work.
- *Hand-Device Distance*: we test the system when the hand waves in different categories of hand-device distance as shown in Fig. 21. AudioGest achieves satisfied accuracy when the distance is below 10 cm (which is the typical using scenario for most users). We also observe its performance decreases when the hand waves in a far distance from the device (the COTS microphone cannot capture the echo-sound due to its capability limitation).
- *Environmental Motion*: as Fig. 17 shows, we test our system under five environmental motion circumstances—Quiet (no audible noise and human motion), Noisy (playing music loudly), Dynamic1 (with human walking back and forth in around 4 meters away the device), Dynamic2 (with human walking back and forth in around 2 meters away) and Dynamic3 (with human walking back and forth nearby, around 0.5 meter). We can see AudioGest works well under first three cases (especially, it is nearly unaffected by human noise).
- *Time Elapse*: we also test its performance under different elapsed time periods—6 hours, 1 day, 3 days, 1 week, and 10 days, without tuning parameters. We conduct a comparison experiment to study the performance of the system adopting and not adopting the proposed signal denoising method (i.e., FFT normalization). As the results shown in Fig. 18, by applying FFT normalization, AudioGest achieves about 35 to 70 percent performance increase when dealing with the signal drifting challenge, which demonstrates the effectiveness of our denoising approach.

7.2 In-Situ Experiments

Figs. 19 and 20 show the system performance in some typical daily-living environments. Two subjects (U1 and U2) participate in the test. We ask the subjects to use three

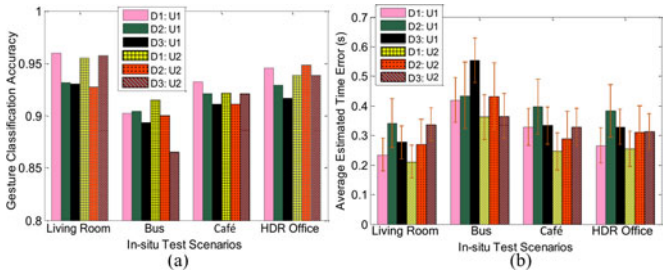


Fig. 19. (a) The average gesture classification accuracy for in-situ test. (b) The average estimation error of hand in-air duration for in-situ test.

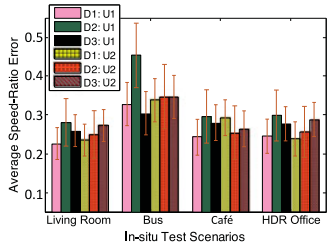


Fig. 20. The average speed-ratio estimation error of hand movement for in-situ test.

mobile devices in a living room ($5\ \mathrm{m} \times 3.5\ \mathrm{m}$), on a bus, in a Cafe, and in an HDR (Higher Degree by Research) space (around $15\ \mathrm{m} \times 10\ \mathrm{m}$, contains > 20 students). We collect 1,200 hand gestures (Living Room: 360, Bus: 240, Cafe: 240, HRD Space: 360). The in-situ testing spans around two weeks upon participants' time availability. Under the living room and HDR office, AudioGest performs similarly to our micro-benchmark since such testing scenarios are usually with less environmental motion inferences. When coming to the bus (the most dynamic environment but also where people usually use the mobile devices), the performance is degraded to an average 89.67 percent accuracy, and the segmentation (i.e., hand in-air duration) and speed-ratio accuracy also decrease, which is mainly caused by the narrow space and unpredictable motion influences on the bus.

7.3 Comparing with the State-of-the-Art

This section compares our AudioGest with seven state-of-the-art HGR systems in terms of detection mechanism, hardware, testing environment, system training requirement and detection capacity/resolution as well as the accuracy, shown in Table 3. Briefly, except for SoundWave [3], other HGR systems mainly exploit *Radio Frequency* (RF) signals to recognize hand motions. Those RF signals are either from COTS or modified WiFi and GSM infrastructures (e.g., WiGest [1], WiSee [10] and SideSwipe [32]), or radars (e.g., FineGesture [9] and RadarGesture [33]), or generated by specialized hardwares (e.g., AllSee [4]). While bearing many advantages, they are either built upon extra hardwares or available WIFI signals, which may be impractical under some circumstances (see discussions in Section 1).

Unlike above HGR systems, SoundWave is one pioneering work to exploit the Doppler effect of sound wave reflected by hands, sharing the same hand gesture recognition mechanism as AudioGest. It mainly adopts a percentage-threshold based dynamic peak tracking method to capture the frequency shifts. Different from SoundWave, our system aims to quantitatively measure the hand waving speed, rang and in-air time (see Fig. 5). More importantly, we provide a mathematical model for interpreting Doppler Effect into hand motion (see Sections 3.1 and 6.4) by linking the equation of Doppler Frequency shift and Newton's law of motion of hand gestures.

We compare our work with SoundWave and other HGR systems in a high-level of view, shown in Table 3. Accurately detecting the frequency shifts is the foundation of HGR systems based on Doppler Effect. Without a good performance in capturing frequency shifts, both our system and Soundwave cannot achieve an accurate hand gesture recognition. To this end, we compare AudioGest with SoundWave by two experimental cases in terms of the performance of detecting frequency shifts. Fig. 21 depicts how SoundWave detects the bandwidth of shifted frequency. When four or more FFT frames (i.e., 2,048-point segmentation) in succession are detected with frequency shifts, SoundWave will consider a hand motion is happened.

Experimental Case 1. Figs. 22b, 22c, and 22d compare the detection results of SoundWave and AudioGest for a *Slow-Speed* clockwise circling case. In Fig. 22a, we observe that the hand is currently moving away from the microphone at $t = 2.38\ s$, and towards the microphone at $t = 3.3133\ s$. However, SoundWave cannot accurately detect frequency shifts in such two FFT frames (see Figs. 22b and 22c) since both the second peaks are less than a threshold 30 percent and the lower point is below 10 percent, thus leading the recognition of "no motion". Fig. 22d shows the result of our method, in which we first utilize *Squared Continuous Frame Subtraction* and *Gaussian Smoothing* to get the shifted frequency area and then transfer it into a hand radial speed

TABLE 3
Comparison of Typical Device-Free Localization Systems

Comparison Items	WiGest [1]	FineGesture [9]	AllSee [4]	SoundWave [3]	SideSwipe [32]	RadarGesture [33]	WiSee [10]	AudioGest
Measured Signal	RSSI	RSS, Phase, CSI	RF signal	Audio	GSM signal	FMCW Radar	OFDM radio	Audio
Need extra hardware?	No	Yes	Yes	No	Yes	Yes	Yes	No
Test in dynamic environment? (e.g., bus)	No	Yes	No	No	No	No	No	Yes
Need training?	No	Yes (kNN)	No	No	Yes (SVM)	No	No	No
Sense gesture contexts? (e.g., speed, range)	Yes (speed)	No	No	No	No	Yes (speed, range)	No	Yes (relative speed & range)
Accuracy	96%	92%	97%	94.5%	87.2%	N/A (hand tracking)	94%	95.1%
Gesture Resolution	36	25	8	5	14	N/A (hand tracking)	9	54 (randomly choose two attributes)

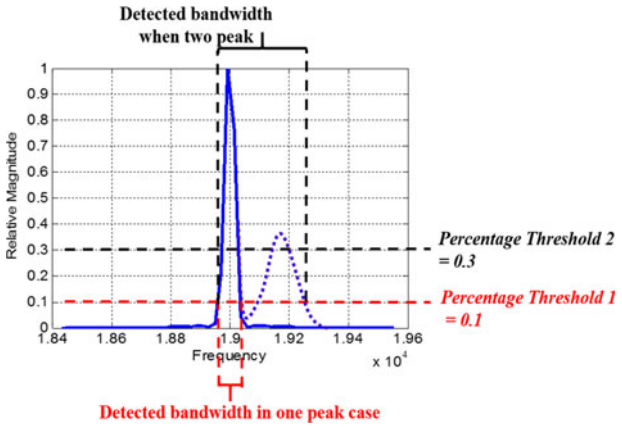


Fig. 21. SoundWave detects the frequency shift based on a percentage-threshold method.

curve. Both frequency shifts as well as hand speed in these two frames are successfully detected and estimated.

Experimental Case 2. Figs. 23b, 23c, and 23d illustrate another detection results for a *Fast-Speed* clockwise circling case. Similarly, although SoundWave can successfully detect the happening of hand motion, it still fails to accurately estimate shifted bandwidth (missing the third peak), which results in incorrect hand speed estimation. Actually, those two FFT frames represent two peak speeds during the hand waving. Fig. 23d shows our result, which correctly quantifies the bandwidth of the shifted frequency and captures the peak speeds.

To summarize, from the perspective of technique and methodology, percentage threshold-based dynamic peak tracking in SoundWave is a promising and efficient method that can deal with the hardware diversities and signal drifts. The FFT Normalization in our paper actually serves the same purpose. However the rest techniques introduced by our system including *Squared Continuous Subtraction*, *Gaussian Smoothing* and *Hand Radial Speed Transformation* make AudioGest free of percentage threshold chosen and more accurate in quantifying shift frequency bandwidth.

8 Discussion

This section will discuss the limitations of our work that are left for the future work.

Separation of the Speaker and Microphone: In AudioGest, we focus on multi-modal hand gesture recognition with only one pair of microphone and speaker. Our system requires that the microphone and speaker are placed in different places. The rationale of speaker-microphone separation lies on *i)* reducing the self-inference from the speaker; *ii)* increasing the performance of microphone; and *iii)* limited deployment space in a mobile device.

Gesture Trajectory: By making sensing of Doppler Effect, AudioGest can recognize six types of pre-defined basic gestures regardless of other hand motion attributes. The starting and stopping points of those gestures are quite flexible. However, it is possible that two different gestures generate a same spectrogram, in which we cannot distinguish these two gestures. This is the reason that AudioGest needs to pre-define the hand moving trajectories.

Noise Disturbance to Human: Considering normal human hearing scope of 55∼18 kHz, AudioGest emits a 19 kHz single tone sound-wave. At the same time, to largely reduce

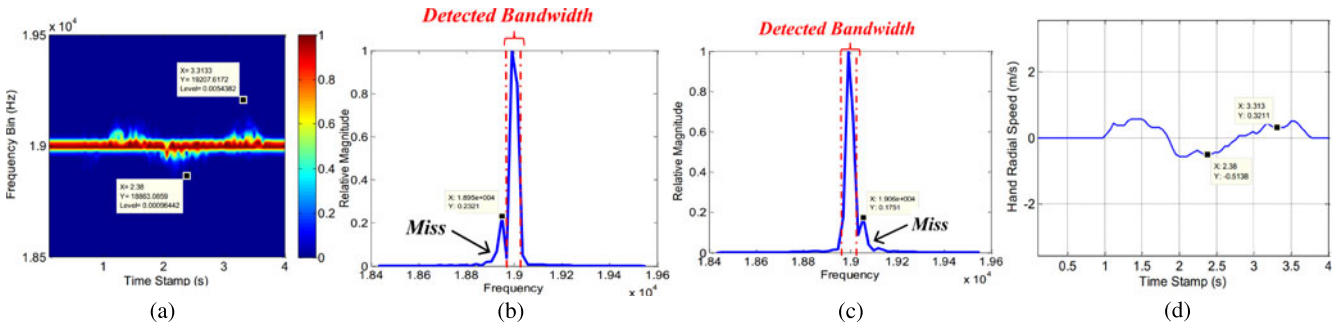


Fig. 22. Experimental case 1: A slow-speed clockwise hand circling. (a) The echo signal's spectrogram after FFT normalization. (b) The detected bandwidth of frequency shift at $t = 2.38$ s (i.e., FFT Frame 51) by SoundWave. (c) The detected bandwidth at $t = 3.3133$ s (i.e., FFT Frame 71) by SoundWave. (d) The real-time hand radial velocity detected by AudioGest.

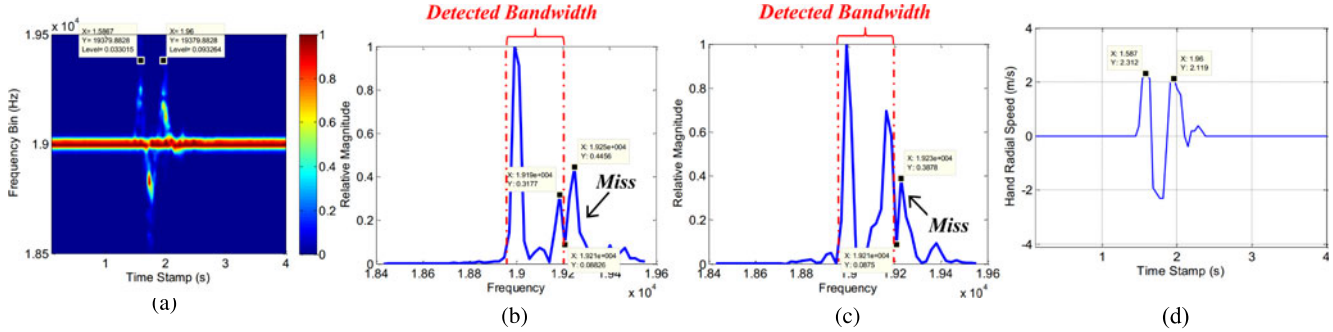


Fig. 23. Experimental case 2: A fast-speed clockwise hand circling. (a) The echo signal's spectrogram after FFT normalization. (b) The detected bandwidth of frequency shift at frame $t = 1.5867$ s by SoundWave. (c) The detected bandwidth at frame $t = 1.96$ s by SoundWave. (d) The real-time hand radial velocity calculated by AudioGest.

the possible disturbance brought by the sound, we adjust the sound volume into a very low intensity since our system aims to detect hand movements around the vicinity of a mobile device. A 13-year-old young female participates in our experiments and she does not feel any uncomfortable while using our HGR system.

Limited Hand Gesture Numbers: AudioGest can provide up to 54 control commands for upper-layer applications by co-recognizing four types of hand motion attributes. It, however, can only distinguish overall 6 hand gestures accurately. In the future, we will investigate this from two ways: *i)* mining other features from the spectrogram of reflected signals to facilitate our physical model in order to recognize more hand gestures; and *ii)* adopting two or more microphones to enable a real-time hand motion tracking.

Dealing with Environment Motion: As the system robustness evaluation shows, AudioGest's performance decreases for some challenging scenarios such as the device orientation greatly changes ($> \pi/4$) and human motions at the vicinity of device (< 0.5 m). However, such issues can be addressed by two possible ways: *i)* exploring the built-in 3-axis accelerometer to detect the orientation of the device, then real-time updating parameters and hand-gesture recognition rules accordingly; *ii)* borrowing the idea from radar to transmit MFSK (multiple frequency shift keying) audio signal, enabling multiple-target range sensing, hence distinguishing the nearby environmental motion and hand movement.

9 Conclusion

To summarize, this paper has shown how one single pair of microphone and speaker can achieve multi-modal hand motion detection. AudioGest thoroughly exploits the Doppler frequency shift from hand movement and accurately interprets the spectrogram of echo signals into the multi-modal hand motion attributes. Our system only uses a single pair of COTS speaker & microphone without any extra hardware, and it is capable of accurately recovering hand's real-time radial velocity, thus decodes the hand moving direction, waving speed, and in-air range. The real-world experiments demonstrate the feasibility and effectiveness of our system, which marks an important step towards enabling accurate and ubiquitous gesture recognition.

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] Y. Ren, Y. Chen, M. C. Chuah, and J. Yang, "User verification leveraging gait recognition for smartphone enabled mobile healthcare systems," *IEEE Trans. Mobile Comput.*, vol. 14, no. 9, pp. 1961–1974, Sep. 2015.

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

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

[5] L. Yang, et al., "Unlocking smart phone through handwaving biometrics," *IEEE Trans. Mobile Comput.*, vol. 14, no. 5, pp. 1044–1055, May 2015.

[6] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," *Commun. ACM*, vol. 54, no. 2, pp. 60–71, 2011.

[7] D. Kim, et al., "Digits: Freehand 3D interactions anywhere using a wrist-worn gloveless sensor," in *Proc. ACM Symp. User Interface Softw. Technol.*, 2012, pp. 167–176.

[8] C. Wang, Z. Liu, and S.-C. Chan, "Superpixel-based hand gesture recognition with kinect depth camera," *IEEE Trans. Multimedia*, vol. 17, no. 1, pp. 29–39, Jan. 2015.

[9] P. Melgarejo, X. Zhang, P. Ramanathan, and D. Chu, "Leveraging directional antenna capabilities for fine-grained gesture recognition," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2014, pp. 541–551.

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

[11] G. Deng and L. Cahill, "An adaptive gaussian filter for noise reduction and edge detection," in *Proc. IEEE Nuclear Sci. Symp. Med. Imag. Conf.*, 1993, pp. 1615–1619.

[12] J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li, "Gesture recognition with a 3-D accelerometer," in *Ubiquitous Intelligence and Computing*. Berlin, Germany: Springer, 2009, pp. 25–38.

[13] G. Cohn, D. Morris, S. Patel, and D. Tan, "Humantenna: Using the body as an antenna for real-time whole-body interaction," in *Proc. ACM SIGCHI Conf. Human Factors Comput. Syst.*, 2012, pp. 1901–1910.

[14] T. Park, J. Lee, I. Hwang, C. Yoo, L. Nachman, and J. Song, "E-gesture: A collaborative architecture for energy-efficient gesture recognition with hand-worn sensor and mobile devices," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, 2011, pp. 260–273.

[15] H. Ketabdar, P. Moghadam, B. Naderi, and M. Roshandel, "Magnetic signatures in air for mobile devices," in *Proc. ACM 14th Int. Conf. Human-Comput. Interaction Mobile Devices Serv. Companion*, 2012, pp. 185–188.

[16] Z. Lu, X. Chen, Q. Li, X. Zhang, and P. Zhou, "A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices," *IEEE Trans. Human-Mach. Syst.*, vol. 44, no. 2, pp. 293–299, Apr. 2014.

[17] Z. Li, R. Robucci, N. Banerjee, and C. Patel, "Tongue-n-cheek: Non-contact tongue gesture recognition," in *Proc. 14th Int. Conf. Inf. Process. Sensor Netw.*, 2015, pp. 95–105.

[18] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes," in *Proc. 20th Annu. ACM Symp. User Interface Softw. Technol.*, 2007, pp. 159–168.

[19] Y. Li, "Protractor: A fast and accurate gesture recognizer," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, 2010, pp. 2169–2172.

[20] T. Starner and A. Pentland, "Real-time american sign language recognition from video using hidden markov models," in *Motion-Based Recognition*. Berlin, Germany: Springer, 1997, pp. 227–243.

[21] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2006, vol. 2, pp. 1521–1527.

[22] N. H. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *IEEE Trans. Instrumentation Meas.*, vol. 60, no. 11, pp. 3592–3607, Nov. 2011.

[23] P. Molchanov, S. Gupta, K. Kim, and K. Pulli, "Multi-sensor system for driver's hand-gesture recognition," in *Proc. IEEE Conf. Workshops Autom. Face Gesture Recog.*, 2015, pp. 1–8.

[24] X. Zhao, X. Li, C. Pang, X. Zhu, and Q. Z. Sheng, "Online human gesture recognition from motion data streams," in *Proc. ACM Int. Conf. Multimedia*, 2013, pp. 23–32.

[25] F. Adib and D. Katabi, "See through walls with wifi!" in *Proc. ACM SIGCOMM Conf.*, 2013, pp. 75–86.

[26] F. Adib, C.-Y. Hsu, H. Mao, D. Katabi, and F. Durand, "Capturing the human figure through a wall," *ACM Trans. Graph.*, vol. 34, no. 6, 2015, Art. no. 219.

[27] L. Yao, et al., "RF-Care: Device-free posture recognition for elderly people using a passive RFID tag array," in *Proc. Intl. Conf. Mobile Ubiquitous Syst.: Comput. Netw. Serv.*, 2015, pp. 120–129.

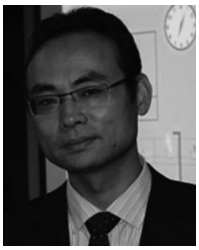
[28] K. Kalgaonkar and B. Raj, "One-handed gesture recognition using ultrasonic doppler sonar," in *Proc. IEEE Int. Conf. Acoustics Speech Signal Process.*, 2009, pp. 1889–1892.

[29] R. Nandakumar, S. Gollakota, and N. Watson, "Contactless sleep apnea detection on smartphones," in *Proc. ACM Int. Conf. Mobile Syst. Appl. Serv.*, 2015, pp. 22–24.

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

[31] S. P. Tarzia, R. P. Dick, P. A. Dinda, and G. Memik, "Sonar-based measurement of user presence and attention," in *Proc. ACM Int. Conf. Ubiquitous Comput.*, 2009, pp. 89–92.
[32] C. Zhao, K.-Y. Chen, M. T. I. Aumi, S. Patel, and M. S. Reynolds, "SideSwipe: detecting in-air gestures around mobile devices using actual gsm signal," in *Proc. ACM Symp. User Interface Softw. Technol.*, 2014, pp. 527–534.
[33] P. Molchanov, S. Gupta, K. Kim, and K. Pulli, "Short-range FMCW monopulse radar for hand-gesture sensing," in *Proc. IEEE Radar Conf.*, 2015, pp. 1491–1496.

Wenjie Ruan received the BS degree in automation and the MSc degree in control science and engineering from the Central South University, China, and the PhD degree from the School of Computer Science, University of Adelaide, Australia. He is a postdoctoral research fellow in the Department of Computer Science, University of Oxford. His research interests include pervasive computing, mobile computing, and data mining. He is a member of the IEEE.



Quan Z. Sheng received the PhD degree in computer science from the University of New South Wales, Sydney, Australia, in 2006. He is a full professor and head of Department of Computing, Macquarie University. His research interests include internet of things, big data analytics, distributed computing, and pervasive computing. He is the recipient of ARC Future Fellowship in 2014, Chris Wallace Award for Outstanding Research Contribution in 2012, and Microsoft Research Fellowship in 2003. He is the author of more than 300 publications. He is a member of the ACM and the IEEE.

Peipei Xu received the BS and the MSc degrees in electronic engineering from the Central South University, China, in 2010 and 2013. She is working toward the PhD degree in the School of Electronic Engineering, UESTC, China. She currently is visiting Australian Centre for Visual Technologies (ACVT), University of Adelaide, Australia. Her research interests include machine learning, mobile computing, tensor decomposition, and its applications. He is a student member of the IEEE.

Lei Yang received the BS and the PhD degrees from the School of Software, Department of Computer Science and Technology, Xian Jiaotong University, respectively. He is the research assistant professor in the Department of Computing, Hong Kong Polytechnic University. Previously, he was a postdoc fellow in the School of Software, Tsinghua University. He is the winner of the Best Paper Award at MobiCom14 and MobiHoc14.

Tao Gu received the BE degree from the Huazhong University of Science and Technology, the MSc degree from Nanyang Technological University, Singapore, and the PhD degree in computer science from the National University of Singapore. He is currently an associate professor in computer science with RMIT University, Australia. His research interests include mobile computing, ubiquitous/pervasive computing, wireless sensor networks, and online social networks. He is a senior member of the IEEE.

Longfei Shangguan received the BE degree from Xidian University, the MS and the PhD degrees from the Hong Kong University of Science and Technology, in 2011, 2013, and 2015, respectively. He is currently a postdoc research associate in the Department of Computer Science, Princeton University. He is a member of the IEEE and the ACM.