AudioGest: Enabling Fine-Grained Hand Gesture Detection by Decoding Echo Signal

Wenjie Ruan
School of Computer Science
The University of Adelaide
Adelaide, Australia
wenjie.ruan@adelaide.edu.au

Quan Z. Sheng
School of Computer Science
The University of Adelaide
Adelaide, Australia
michael.sheng@adelaide.edu.au

Lei Yang
School of Software
Tsinghua University
Beijing, China
young@tagsys.org

Tao Gu
School of CS & IT
RMIT University
Melbourne, Australia
tao.gu@rmit.edu.au

Peipei Xu
School of Electronic
Engineering, UESTC
Chengdu, China
peipei.xu6@gmail.com

Longfei Shangguan
Computer Science Department
Princeton University
Princeton, USA
longfeis@cs.princeton.edu

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, 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 fine-grained hand detection. Our system is able to accurately recognize various hand gestures, estimate the hand in-air time, as well as 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 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 implement the proof-of-concept prototype in three different electronic devices and extensively evaluate the system in four real-world scenarios using 3,900 hand gestures that collected by five users for more than two weeks. Our results show that AudioGest can detect six hand gestures with an accuracy up to 96%, and by distinguishing the gesture attributions, it can provide up to 162 control commands for various applications.

Author Keywords

Hand Gestures; Audio; Microphone; FFT; Doppler Effect

UbiComp '16, September 12-16, 2016, Heidelberg, Germany

DOI: http://dx.doi.org/10.1145/2971648.2971736

ACM Classification Keywords

C.3. Special-purpose and Application-based Systems: Miscellaneous

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 has aroused enormous attentions from both industry [2, 3] and academia [4, 21]. Hence many companies and researchers intend to integrate the hand-gesture recognition into our daily devices, including laptops [13], tablets [15], smart phones [7], and gaming consoles [3, 1]. However, a crucial prerequisite of such applications is that the device can accurately and robustly detect gestures at anytime (*e.g.,* poor light condition at night), at anywhere (*e.g.,* in rural area without wireless connection) in a device-free manner (*e.g.,* no need to wear an extra device/sensor) [8, 15, 26].

To tackle such challenging requirements, many state-of-the-art gesture approaches have been developed over last decades such as computer vision [31], inertial sensors [17], ultrasonic sensors [13], infrared sensors [1], depth sensors[3], etc. While promising, most of these systems, however, can only partially satisfy aforementioned requirements [4], such as sensitivity to the light condition (*e.g.,* vision-based methods), being limited for specific applications (*e.g.,*. Leap Motion), high installation and instrumentation overhead (*e.g.,* Kinect), or needing user to wear additional devices/sensors (*e.g.,* wearable sensor based techniques).

As a result, many WiFi-based attempts have recently been proposed to help overcome the above limitations. For example, WiGest [4] exploits the influence of in-air hand movement on the wireless signal strength received by the device from an access point (AP) to recognize the performed gestures. Melgarejo *et al.* [18] 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 [25] exploits the doppler shift in narrow bands extracted from wideband OFDM transmissions to recognize nine different human gestures. Although WiFi-based systems can work under any lighting conditions, and do not require dedicated hardware modification, such 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 tram/bus or traveling in a rural area.

To tackle these limitations, we hence introduce AudioGest, a device-free system that can transform the consumer device into an active sonar system by utilizing the microphone and speaker that are already embedded in the device. Compared to other audio-based systems, AudioGest is able to only use one pair of built-in speaker and microphone, with no modification, and no training to achieve fine-grained hand gesture detection. More importantly, our system can also accurately estimate the hand in-air time, average waving speed as well as hand moving range.

Implementing such a practical system, however, requires addressing a number of challenges. First, the ambient noise (*e.g.,* human conversation, electronic noise etc.) dominates the recorded audio signal (see the experiments in Sec. **Weak Echo Signal**). It is hence difficult for us 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 Sec. **Audio Signal Drift**). Since we emit a high-frequency audio signal ($> 18kHz$, making it inaudible to human), the Operational Amplifier (OA) in microphone and speaker both experience severe attenuation, making the magnitude of recorded echoes extremely unstable. Moreover, different microphones/speakers have various OA attenuations, also resulting in signal drifting.

To address such issues, in AudioGest, we propose three main techniques to tackle the aforementioned challenges. First, we introduce a FFT-based normalization that substantially adjusts the magnitude of FFT frequency bin in different timestamps to a same level, removing the influence of OA attenuation in high-frequency part (see details in Sec. **FFT Normalization**). Then, we conduct a *Squared Continuous Frame Subtraction*, in which we first subtract the spectrum of current audio frame by previous frame and then square the magnitudes of frequency bins, further eliminating the nearby human motion influence (see details in Sec. **Squared Continuous Frame Subtraction**). Furthermore, we utilize a Gaussian smoothing filter [11] to transfer the discrete shifted frequency bins into a contouring area. Then we decode it into the real-time hand moving velocity curve based on the Doppler frequency shift (see details in Sec. **Transforming Frequency Shift Area into Velocity**). Finally, according to such velocity curve, we estimate the hand gesture, moving speed as well as the waving range (see details in Sec. **Gesture Recognition**). 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-suit experiments with five users for more than two weeks demonstrate the feasibility and accuracy of AudioGest in various living environments.
- We propose a denoising pipeline that can not only abstract the Doppler frequency shifts from weak echo signals, but also deal 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 average 95.1% accuracy, as well as precisely distinguish the magnitude differences of various hand speed and moving range, being able to provide up to 162 control commands.

RELATED WORK

Prior gesture-recognition system can be categorized into two general types: *wearable sensor/device based* gesture recognition and *device-free* gesture recognition.

Wearable Devices based Gesture Recognition: Wearable sensor/device based systems utilize various sensors (*e.g.,* 3-axis accelerometer [33, 2], inertial sensor [9], gyroscope [7] or other smart devices [8, 24]) to sense the movement of hand or arm. For example, some researchers infer the hand movement by wearing a shaped magnet [16]. Humantenna [9] requires the user to wear a small Wireless Data Acquisition Unit enabling the human body as an antenna for sensing whole-body gestures. With the advanced build-in sensors in mobile device, the system in [7] transfer the acceleration recorded by smartphone into a real-time hand moving trajectory. All these systems, however, require the tracked subjects to carry a device/sensor, which might be impractical for some applications (e.g., old people with dementia).

Device-free Gesture Recognition: This category can be further classified into vision-based, environmental sensor based and RF-based as well as sonar-based approaches. Video-based hand-gesture recognition systems often do the hand-region segmentation using color and/or depth information, and sequences of features for dynamic gestures are used to train classifiers, such as Hidden Markov Models (HMM) [29], conditional random fields [32], SVM [10], DNN [19]. However, vision-based techniques are usually regarded as being privacy-invasive. They also require users within the LOS (line of sight) of cameras, and fail to work in dimmed environments, and incur high computational cost. Also, some commercialized hand recognition systems have been emerged lately, such as Leap Motion [1] that explores multiple channels of reflected infrared signals to identify hand gestures, and Kinect [3] that uses depth sensor to enable in-air 3D skeleton tracking.

Recently, RF-based gesture recognition systems are also very popular due to its low-cost and being less intrusive [12, 4, 28]. For example, WiVi [6, 5] uses ISAR technique to track the RF beam, enabling a through-wall gesture recognition. RF-Care [34, 35, 27] proposes to recognize human gestures and activities in a device-free manner based on a passive RFID

(Radio-frequency identification) array. WiSee [25] can exploit the doppler shift in narrow bands in wide-band OFDM (Orthogonal Frequency Division Multiplexing) transmissions to recognize 9 different human gestures. WiGest [4] explores the effect of the in-air hand motion on the RSSI in WiFi to infer the hand moving directions as well as speeds. Melgarejo *et al.* [18] leverage the directional antenna and short-range wireless propagation properties to recognize 25 standard American Sign Language gestures. AllSee [15] designs a very power-efficient hardware that extracts gesture information from existing wireless signals.

SonarGest [14] is one of the pioneering audio-based hand recognition systems (HRG), which uses three ultrasonic receivers and one transmitter to recognize 8 hand gestures. The technique utilized is a supervised Gaussian Mixture Model that can capture the distribution of the feature vectors obtained from the Doppler signal of gestures. However, it needs to collect training data (potentially labour-intensive and time-consuming) and requires extra sonic hardware. SoundWave [13] is another pioneering HRG system by exploiting audio Doppler effect as well. It only utilizes the built-in speakers and microphones in computers and require no training. SoundWave designs a threshold-based dynamic peak tracking technique to effectively capture the Doppler shifts, thus can distinguish five different hand gestures. Most recently, researchers are trying to transform the COTS speaker and microphone into a sonar system to detect human breath [22], track a finger movement [23], sense user's presence [30], etc. 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 microphone. However, such systems either need two microphones or require design specialized sound-wave that is power-intensive. Motivated by, but different to, the previous works, this paper only utilizes one speaker and one microphone by emitting single-tone audio to achieve a multi-level gesture recognition. It can also decode the echo's spectrogram into real-time hand waving velocity by thoroughly exploring the relations between hand motion and observed frequency shifts of the echo signal.

PRELIMINARIES

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. Such system, however, needs the support of high-definition audio capabilities. Fortunately, mobile device hardware is increasingly supporting high-definition audio capabilities, which can support up to 22kHz response frequency and typical 44.1kHz or 48kHz sampling rate, enabling the possibility of fine-grained hand detection.

Our system is motivated by a prevalent law in the physical world, namely *Doppler Effect*. Doppler effect illustrates and quantifies the wavelength changes when wave energy of sound or radio waves travels between two objects if one or both of them are moving. The Doppler effect causes the received frequency of a source to differ from the sent frequency if

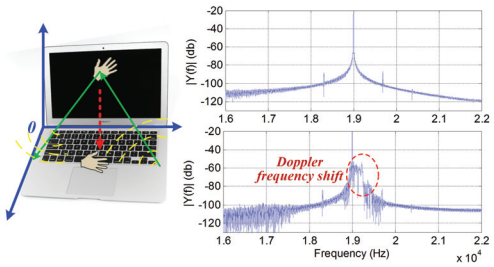


Figure 1: Illustration of Doppler Frequency Shift

there is motion that is increasing or decreasing the distance between the source and the receiver. 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) moves in the air. Hence, though most of sound waves stay unchanged, part of acoustic waves that are reflected by a moving hand will experience a Doppler frequency shift as measured by Eqn. 1.

$$f_{received} = \frac{1 + v_{rad}/v_{sound}}{1 - v_{rad}/v_{sound}} f_{sound} \quad (1)$$

where v_{rad} means the radical speed of hand to microphone. As Fig. 1 shows, when a hand moving 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, as well as to estimate the moving speed and duration of a hand in air.

EMPIRICAL STUDIES AND CHALLENGES

In this section, we will conduct some empirical studies and identify the challenges that we need to deal with.

Weak Echo Signal

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

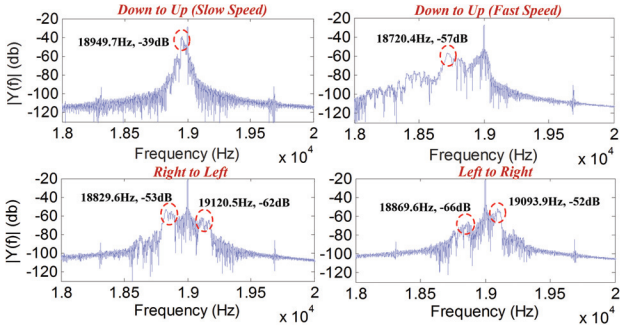


Figure 2: The Doppler frequency shifts caused by different hand gestures and waving speeds

From Fig. 2, we 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. Thus, how to abstract such weak, vulnerable frequency-bin changes from wide-band[1] audio signals is a big challenge. Moreover, we intend to decode the fine-grained hand moving speed, in-air duration and motion range from such weak echoes. At the same time, the ambient noises (such as human conversation, electronic noise) further increase its difficulty. We will illustrate our solution in Sec. **Audio Signal Segmentation**.

Audio Signal Drift

Another challenge we need to tackle is the audio signal drifts caused by the time elapse and device diversity.

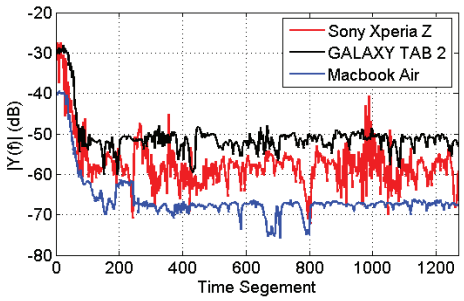


Figure 3: The sound signal drifts for different mobile devices at different time slots

Shown in Fig. 3, we transmit a 19kHz audio signal and record it using three different mobile devices.

Obviously, the recorded audio signal shows fluctuated intensities for the same frequency (especially, the mobile phone exhibits a stronger signal drift). Such signal excursion will greatly hinder the system's scalability, which means a method that works well in one device may be incapable for other devices or after several days. We will deal with this challenge in Sec. **FFT Normalization**.

SYSTEM CONCEPTUAL OVERVIEW

Fig. 4 shows the system architecture of AudioGest, including three conceptual layers, namely the *gesture detection* layer, the *gesture categorization* layer and the *application* layer. The gesture detection layer is the key part of the whole system, which outputs four kinds of gesture contexts - waving direction, hand's average speed and in-air duration, as well as waving range. The gesture categorization layer categorizes different basic gesture characteristics from previous layer into different semantics. 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 [4, 15], AudioGest provides four types of hand gesture-contexts. By combination, it can theoretically provide up to $6 \times 3 \times 3 \times 3 = 162$ control commands, which we thus called *fine-grained* hand gesture recognition. It is noted that AudioGest can support a smaller categorization (*e.g.,* classify the in-air duration into four or five levels) however it may deteriorate the detection accuracy. Vice-versa, we can use a coarse-grained categorization to increase the estimation accuracy. For example, for an e-book App, which 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 more flexibility to the application layer. Finally, the application layer maps different gestures to control commands for various applications.

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

THE AUDIOGEST SYSTEM

We first introduce how to design the transmitted audio signal. Human normal audible scope is 20Hz∼18kHz. To avoid annoying human audibility, under no circumstance, should AudioGest produce the sound signal below 18kHz (to be more safe, we make it 18.5kHz). Assuming that the fastest hand moving speed is $4m/s$ [13], then the largest Doppler frequency shift[2] $\Delta f_{doppler} = (2v_{hand}/v_{sound})f_{transmit} = 470.6Hz$. Hence, if the mobile device transmits a 19kHz sound, then the received audio signal is 18,529.4Hz∼19,470.6Hz, satisfying the requirement. Also, we save a bandwidth ($2\Delta f_{doppler} = 941.2Hz$) for another possible audio channel[3]. Although microphones in some devices can support a 48kHz or even 192kHz sampling rate, we adopt a more general 44.1kHz sampling rate.

FFT Normalization

As aforementioned, the raw data recorded by microphones not only contain audible noise but also introduce the signal drifts due to temporal changes and diverse hardwares. This section introduces a FFT-based normalization to deal with such issues.

Since our targeted sound frequency band is 18.5kHz∼19.5kHz, we will only do analysis to audio signals within this narrow bandwidth in the following processing. Such processing will naturally filter out the influence of audible noise without adding an extra band-pass filter. Then, we adopt a 2048-point hamming window to segment the filtered signal into audio frames[4], apply a 2048-point FFT [5] to each frame get the sound spectrogram, showing as the left graph in Fig. 5. 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 overall 3,600 seconds 19kHz sound signal using three different mobile devices (*i.e.,* MacBook Air laptop, Sumsung Galaxy S4 smartphone

[2]Since we do not know the transmitted sound frequency beforehand, we use a larger possible transmitted frequency 20kHz, $v_{sound} = 340m/s$ under 15 $^{\circ}C$.

[3]It means we can use another speaker channel to transmit a 20kHz sound, and the received signal is 19,529.4Hz∼20,470.6Hz, which lies in the recording capability of a microphone but without inference with another speaker channel.

[4]Each frame represents $2,048/44,100 = 0.0464s$ audio signal.

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

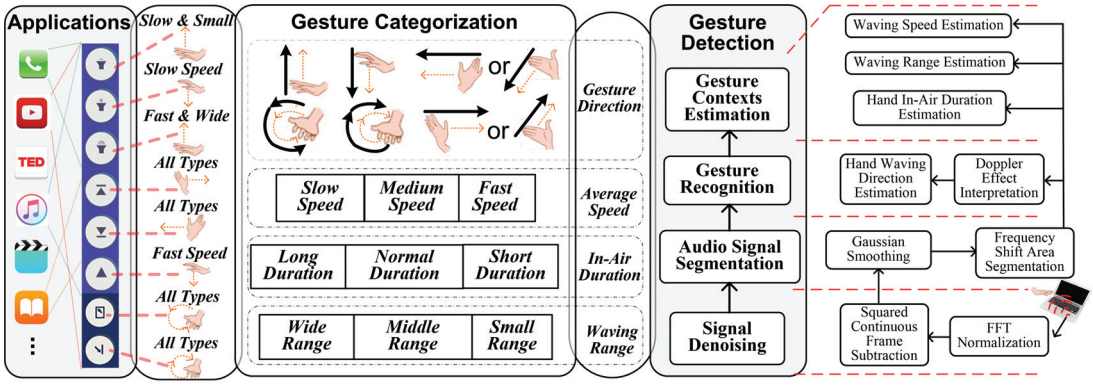


Figure 4: Overview of the system for hand gesture detection

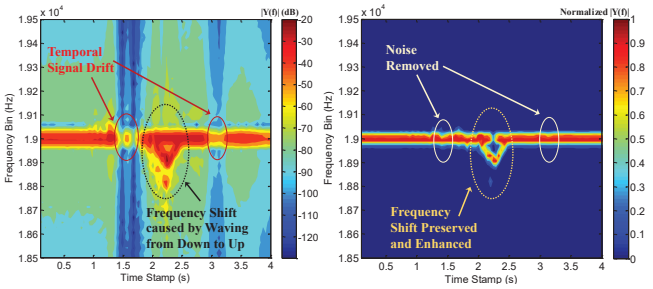


Figure 5: Left: raw audio spectrogram; Right: audio spectrogram after FFT normalization

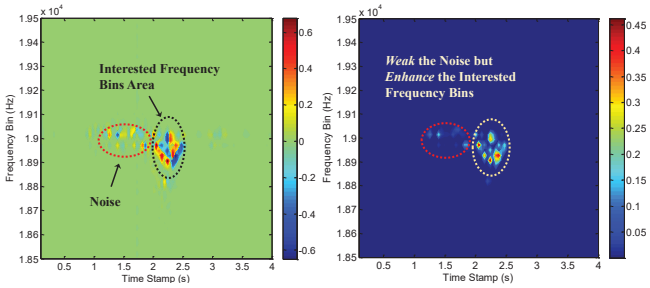


Figure 6: Left: the spectrogram after continuous frame subtraction; Right: the spectrogram after the square calculation

and Sumsung Tab-2 tablet) and then segment the signal into frames of 2,048-point length. We can observe that, although the magnitude of the frequency bins for different frames show unpredictable signal excursions (*e.g.,* the magnitude in 19kHz bin spans from -83dB∼-24dB), the relative magnitudes for every single sound frame exhibit stable and robust to the time-elapse and device diversity (*i.e.,* each spectrum shows a similar shape). It is noted that we intend to perceive the Doppler frequency shifts to infer hand gestures. Hence we are more concerned about how the peak frequency bin changes along 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 as the right graph in Fig. 5, 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.

Audio Signal Segmentation

Squared Continuous Frame Subtraction

To perceive the magnitude changes of frequency bins, we further conduct a *Squared Continuous Frame Subtraction*, in 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. 6 (*i.e.,* remove the unchanged 19kHz bin in Fig. 5 and highlight the changed frequency bins). The square calculation will further enhance the frequency-bin changes caused by hand's movement but weaken the bins due to the noise (see the right graph in Fig. 6, 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.

Gaussian Smoothing

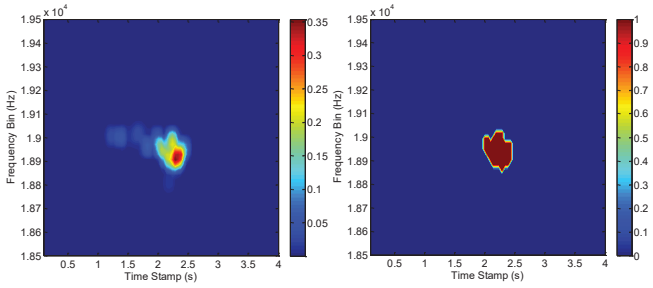


Figure 7: Left: the spectrogram after Gaussian Smooth Filter; Right: the segmented area where Doppler Frequency shift happens

Revisit the right graph in Fig. 6, intuitively, we can view such spectrogram graph as an image, then what we are interested is to connect those pixels and augment it into a zone. Hence, 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 a image.

For our two-dimension image, the following function is used for smoothing:

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

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.

As the left graph in Fig. 7 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. As shown in the right graph of Fig. 7, we can successfully segment the frequency zone that Doppler shift happens.

Fig. 8 depicts the spectrograms after denoising and our segmentation results for various hand gestures waving with different speeds. In the next, we need to accurately interpret such segmented frequency shifts into different hand gestures, as well as estimate the gesture in-air duration, hand waving speed and range.

Doppler Effect Interpretation

In this section, we first choose two typical hand gestures to do the Doppler effect interpretation, showing the relation between audio spectrograms and hand gestures. From Eqn. 1, since $v_{sound} \gg v_{rad}$, we have

$$\Delta f = \frac{2f_{sound}v_{rad}}{v_{sound}} \quad (3)$$

where $\Delta f = f_{received} - f_{sound}$. As Fig. 9 shows, assuming that hand moving path has θ_{hand} with the microphone and the hand moving speed is v_{hand}, we have

$$v_{rad} = v_{hand}\cos\theta_{hand} \quad (4)$$

Furthermore, we can derive the relation based on Eqn. 3 and Eqn. 4 as follows.

$$\Delta f = \frac{2f_{sound}v_{hand}\cos\theta_{hand}}{v_{sound}} \propto v_{hand}\cos\theta_{hand} \quad (5)$$

We take two examples[6] to interpret Eqn. 5, showing how we link real-time hand moving gesture with the audio spectrogram. As Fig. 9 depicts, when the hand moving from *Right to Left*, θ_{hand} gradually increases (*e.g.,* from $\pi/6$ to $\pi/2$ then to $2\pi/3$), hence $\cos\theta_{hand}$ decreases[7] 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 19kHz) to zero, then to low-frequency (*i.e.,* lower than 19kHz). 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 $\pi/3 \to 0 \to \pi/2 \to \pi \to \pi/3$ the right graph of Fig. 9). Thus, the audio frequency shifts towards high-frequency at first, then goes back to 19kHz, further moves to the low-frequency, then it goes back to zero, continuously moves to high-frequency[8].

Transforming Frequency Shift Area into Hand Velocity

Based on Eqn. 5, we can model the frequency shift with real-time hand radical velocity as

$$\begin{aligned} f_{received}(t) - f_{sound} &= \frac{2f_{sound}}{v_{sound}}v_{hand}(t)\cos\theta_{hand}(t) \\ &= \frac{2}{\lambda_{sound}}v_{rad}(t) \end{aligned} \quad (6)$$

Furthermore, we can derive hand radical velocity $v_{rad}(t) = 0.5\lambda_{sound}(f_{shift}(t) - f_{sound})$.

As the left graph of Fig. 10 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 radical velocity of hand as shown in the right top graph in Fig. 10. Essentially, the sign of hand radical 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.

Gesture Recognition

In this section, we introduce in details on how we estimate the hand waving direction, speed and in-air duration as well as moving range given the hand radical velocity curve.

Recognizing the Waving Direction

In Sec. **Doppler Effect Interpretation**, we show that how we link the hand moving directions with its generated audio spectrogram. Similarly, based on the direction changes of radical velocity (*i.e.,* whether its value is negative or positive, which is 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 angle category. Based on a sequence of angle categories and its durations, we can further detect different gesture types. Fig. 8 shows some examples, if the angle range is only in $[0,\pi/2]$, then the hand moves from *up to down*; if the angle range is only in $[\pi/2,\pi]$, then the hand moves from *down to up*; if the angle is $[0,\pi/2] \to [\pi/2,\pi]$ and the time duration in $[0,\pi/2]$ is longer than in $[\pi/2,\pi]$, then the hand moves from *right to left*; if the angle is $[0,\pi/2] \to [\pi/2,\pi]$ but the time duration in $[0,\pi/2]$ is shorter than in $[\pi/2,\pi]$, then the hand moves from *left to right*; if the angle is from $[\pi/2,\pi] \to [0,\pi/2] \to [\pi/2,\pi]$, then the hand moves in *anti-clockwise circle*; if the angle is $[0,\pi/2] \to [\pi/2,\pi] \to [0,\pi/2]$, then the hand moves in *clockwise circle*. Worthing to mention, unlike most of current hand-gestures recognition systems that highly depend on semi-supervised/supervised machine learning methods [26], our proposed method is originated from the interpretation of *Doppler Effect*, hence no need to collect

[6] We choose two typical but more complicated gestures to do the interpretation.

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

[8] Based on Eqn. 5, Δf actually is determined by both v_{hand} and $\cos\theta_{hand}$. And v_{hand} represents the hand speed (a nonnegative scalar), being zero at starting and ending point of hand moving, hence $\cos\theta_{hand}$ (ranging between -1 to 1, and traversing 0 multiple times) dominates the frequency shift.

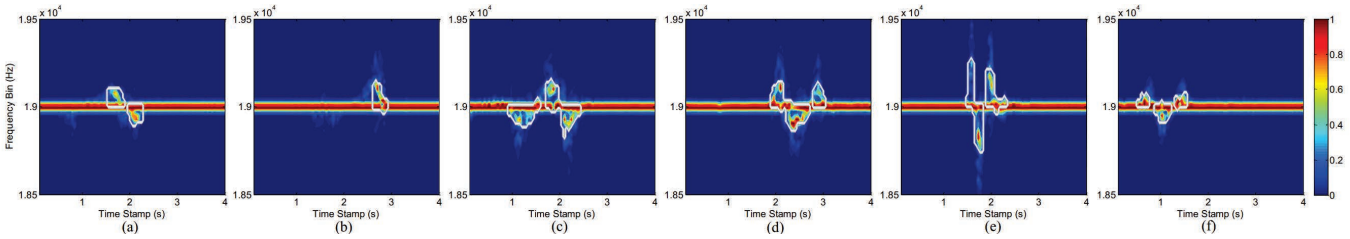


Figure 8: The denoised spectrograms of different hand gestures with various speeds and their segmentation results: from left to right - waving hand (a) from Right to Left; (b) from Up to Down; (c) Anticlockwise circle; (d) clockwise circle; (e) clockwise circle with fast speed; (f) clockwise circle with slow speed

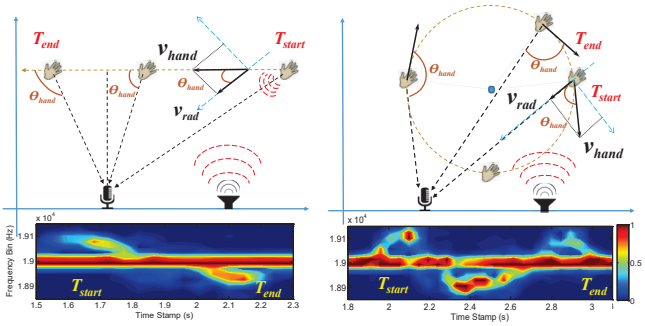


Figure 9: The hand moving path with its generated audio spectrogram. Left: hand moving from Right to Left; Right: hand moving along Clockwise Circle

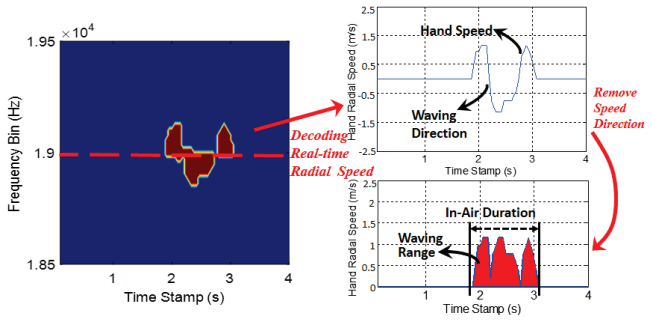


Figure 10: The illustration of transforming frequency shifts into hand velocity, in-air duration and waving range

labeled training data, nor requires to train a classifier. In the next, we will further introduce how we evaluate the speed and range of hand movement.

Estimating Waving Duration and Speed

For estimating the hand in-air duration, we can directly measure the time interval that hand radical velocity is not equal to zero (*e.g.,* the time length marked by dot-line in Fig. 10). Then remaining problem is how we measure the average hand moving speed. Please note that the velocity curve we estimate is the hand radical speed (towards the microphone) instead of the real hand moving speed that we interested in[9]. In this paper, as aforementioned, 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 the same gesture type, we want to evaluate if the hand moving is in slow, medium or fast speed (see Fig. 4).

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. 10. We can observe that, for the same gesture with different speeds, the θ_{hand} actually experiences a same angle range (*e.g.,* $\pi/6 \rightarrow ... \rightarrow \pi/2 \rightarrow ... \rightarrow 2\pi/3$: moving from right to left as in the left graph of Fig. 9) but in different timestamps. As a result, according to Eqn. 4, we can infer that $E(V^1_{hand}) > E(V^2_{hand}) \iff E(V^1_{rad}) > E(V^2_{rad})$, where $V^1_{rad} = \{v^1_{rad}(t_1), v^1_{rad}(t_2), ...\}$ represents the first sequence of hand radical speed we estimated, V^2_{rad} indicates the second sequence of hand radical speed[10]. 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 second using a 2048-point frame), the hand waving duration is $t_{waving} = nT$, then we can calculate the *speed-ration* as

$$S_{ratio} = \frac{E(v_{rad}(t))}{E(v^0_{rad}(t))} = \frac{\frac{1}{n}\sum_{i=1}^{n} v_{rad}(iT)}{E(v^0_{rad}(t))} \quad (7)$$

where $E(*)$ means expectation or mean value; $v^0_{rad}(t)$ represents a baseline of the hand moving speeds and we assume $E(v^0_{rad}(t)) = 1$ for simplicity[11]. Hence, we have $S_{ratio} = \frac{1}{n}\sum_{i=1}^{n} v_{rad}(iT)$, namely the mean value of our estimated radical-speed. Intuitively, a bigger S_{ratio} represents a faster hand movement.

Estimating Waving Range

Similar to the waving speed, we cannot estimate exactly how much distance the hand moves using one microphone. By inheriting the idea in evaluating the waving speed, we also define a *range-ratio* to measure the relative magnitude of hand

[9]Theoretically, with a single microphone, we cannot estimate the moving velocity of hand since we cannot accurately measure the angle between hand and microphone. To do so, we at least need two microphones which will leave to our future work.

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

[11]We can definitely find a certain hand waving meets such requirement.

waving range.

$$R_{ratio} = \frac{R_{rad}}{R^0_{rad}} = \frac{\sum_{i=1}^{n} T v_{rad}(iT)}{R^0_{rad}} = \frac{nTS_{ratio}}{R^0_{rad}} \quad (8)$$

where R^0_{rad} represents the baseline of hand waving range that we assume its value 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. 10), where n and S_{ratio} is the hand in-air duration and speed-ratio we estimated.

EXPERIMENTAL SETUP

We have five participants to conduct the testing on three typical mobile devices - laptop, tablet and mobile phone. Specifically, we choose MacBook Air laptop (Intel i5-4250U 1.3GHz, 4GB RAM, 128 SSD, MacOS X 10.11.3), GALAXY Tab-2 tablet and GALAXY S4 smartphone to conduct the experiments without adding any extra hardwares. We name the three devices as *D1*, *D2* and *D3* respectively for simplicity.

Hardware: For the MacBook Air laptop, we run our AudioGest system on the computer using Audio System Toolbox[12] that enables real-time audio signal processing and analysis in MATLAB2015b and Simulink. It also provides the low-latency connectivity for streaming audio from and to sound cards via the Core Audio[13] standard. For the GALAXY tablet and smart-phone, we design the AudioGest system in the Simulink8.6 that provides a library of Simulink blocks for accessing the devices speaker and microphone[14].

Testing Participants: Overall five participants join the experiments. AudioGest decodes the hand gesture via analyzing the reflected audio signal from hand so the hand size influences the testing results. Intuitively, a bigger hand generates a stronger echo signal. Thus we measure the hand size of each participant, listed in Fig. 11. We also mark the five users as *U1*, *U2*, *U3*, *U4* and *U5* respectively.

Ground Truth Collection: As Fig. 12 shows, we use the 3-axis MEMS accelerometer in a smart-watch for collecting ground truth. Generally, the 3-axis accelerometer records acceleration readings along three orthogonal axises. We set the sampling rate 24Hz that is same to our AudioGest system. In this paper, we decode two types of hand gestures: *i)* linear movement, such as waving from *up to down* or *left to right etc.*; *ii)* circle movement, such as waving in *clockwise circle* or *anticlockwise circle*. For the first case, we measure the acceleration of the corresponding direction (remove the gravity if in z-axis, same goes the followings) to calculate the hand in-air time, average hand speed (*i.e.,* $\bar{v} = 1/2at$) and waving range (*i.e.,* $r = 1/2at^2$), then we set a same baseline of waving speed and range as AudioGest to calculate the speed-ratio and range-ratio. For the second case, we keep the hand downward and do the circling movement. Then we can estimate the total acceleration based on the recorded 3 accelerations (*i.e.,* $a_{total} = \sqrt{a_x^2 + a_y^2 + a_z^2}$), and conduct the same calculation to get the ground truth.

EVALUATION

We start with micro-benchmark experiments in a testbed environment at the lab, then we conduct the in-suit tests in four real-world places - Living Room, Bus, Cafe and HDR Office.

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[15], hence we test overall 2,700 hand gestures by collecting around 4.52 minutes audio data.

Gesture Recognition: Fig. 13 shows the gesture classification accuracies of five users for three devices. AudioGest achieves 94.15% gesture type recognition accuracy. In particular, subject U5 can get average 95% accuracy, but U1 achieves 90.15% mean accuracy using the tablet. From its confusion matrix (shown in Fig. 14), 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 enables *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.

Waving Context Estimation: Fig. 15-17 shows the results of estimation errors[16] 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 0.255s 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 the gesture classification and the context estimation, which mainly lie in the fact that U5 has a slightly bigger hand size, which enhances the audio signal reflection.

Parameters Chosen: Fig. 18-19 illustrates how three key parameters influence the performance of our system. The parameter *H-size* specifies the number of rows and columns we

[12] mathworks.com/hardware-support/audio-ast.html

[13] developer.apple.com/library/mac/documentation/MusicAudio/ Conceptual/CoreAudioOverview/

[14] mathworks.com/hardware-support/android-programming-simulink.html

[15] The participants can freely wave with any speed or range, but have to be within the category of defined gesture types. The collection time spans over two weeks based on their available time. We also require the minimum time-interval of two hand gestures is $> 1s$.

[16] Namely, the distance between estimated value with the ground truth (≥ 0).

User ID	Gender	Age	Hand Length	Hand Width
User 1 (U1)	Male	29	17.1 cm	9.2 cm
User 2 (U2)	Female	29	16.4 cm	8.5 cm
User 3 (U3)	Male	27	18.5 cm	10 cm
User 4 (U4)	Female	13	14.7 cm	7.5 cm
User 5 (U5)	Male	23	17.4 cm	9.5 cm

Figure 11: Participants' information

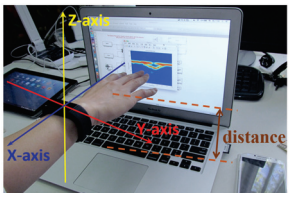


Figure 12: The three-axis accelerometer in smartwatch

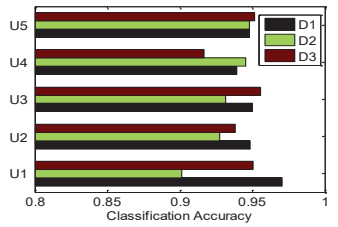


Figure 13: The average gesture classification accuracy for different mobile devices and users

Actual Gestures \ Classified Gestures	Up-Down	Down-Up	Right-Left/ Front-Behind	Left-Right/ Behind-Front	Clock-wise	AntiClockwise	Non-Action
Up-Down	434	0	5	2	1	0	8
Down-Up	0	430	8	9	1	2	0
Right-Left / Front-Behind	3	1	396	23	7	8	12
Left-Right / Behind-Front	1	4	19	411	0	4	11
Clockwise	3	2	6	13	426	0	0
Anti-Clockwise	2	1	8	14	0	425	0

Figure 14: The Confusion Matrix for the gesture classification

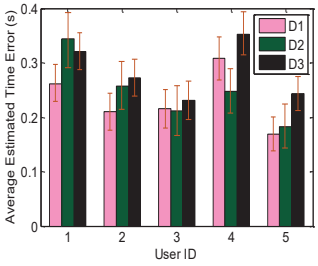


Figure 15: The hand in-air duration estimation error for different mobile devices and users

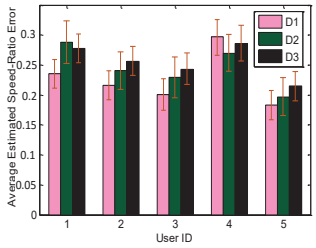


Figure 16: The average speed-ratio estimation error of hand moving for different mobile devices and users

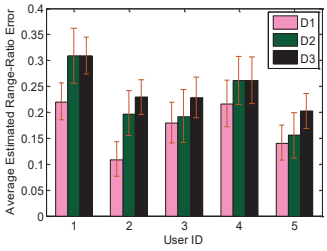


Figure 17: The average range-ratio estimation error of hand moving for different users

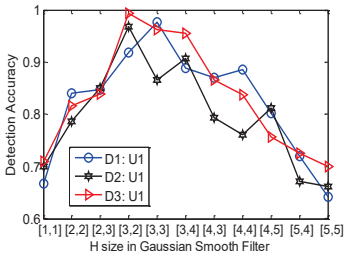


Figure 18: The gesture detection accuracy with parameter *H-size*

use in the gaussian filter (*i.e.,* $H_{size} = [x, y]$ in Eqn. 2). We test overall 11 different H-size when $[x = 3, y = 2]$ performs better. The another 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 system. We can see that the higher the value is, the more both true detection and false detection rates decrease. Hence we choose *Threshold = 0.16* to balance such two detection rates.

System Robustness: We evaluate our system robustness in four ways: *i)* Orientation Angle: as Fig. 20 shows[17], AudioGest performs well when the orientation angle is less than $\pi/4$. Under a $\pi/2$ circumstance, its accuracy greatly decreases to around 60%, which we will leave for further work. *ii)* Hand-Device Distance: when hand waves, we test the system when the hand waves in different categories of hand-device distance[18]. Our system achieves satisfied accuracy when the distance is below 10*cm* (which is the most popular using scenario for most users). But we also observe its performance decreases when the hand waving in a far distance from the device (the COTS microphone cannot capture the echo-sound due to the its capability limitation). *iii)* Environmental Motion: as Fig. 22 shows, we test system performance under five environmental motion circumstances - Quiet (no audible noise and human motion), Noise (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 meters). As we can see, AudioGest works well under first three cases (especially, it is nearly unaffected by human noise). We also test its performance under different elapsed time without tuning the parameters. Thanks to our denoising operation, AudioGest can perfectly deals with the signal drifting challenge. In summary, AudioGest performs accurately under normal circumstance, especially robust to the human noise and signal drifting issue.

[17]We mainly test D2 and D3 from 0 to $\pi/2$, since laptop normally lie flat on the surface.

[18]It is difficult for us to accurately control/measure how hand close to the device while waving, but control the lowest hand-device distance into a range is possible.

In-suit Experiments

Fig. 23-25 illustrate the system performance in some typical daily-living environments. Two subjects (U1 and U2) participate the test. We require the subjects use three mobile devices in a living room ($5m \times 3.5m$), on a bus (when have a seat), Cafe and HDR (Higher Degree by Research) Office (around $15m \times 10m$, contains > 20 students). We collect in total 1,200 hand gestures (Living Room: 360, Bus: 240, Cafe: 240, HRD Office: 360). The in-suit testing spans around two weeks upon participants' time availability. Overall, 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 dynamical environment but also where people usually use the mobile devices), the performance of AudieGest is degraded to an average 89.67% in accuracy, and the segmentation (*i.e.,* hand in-air duration) and the speed-ratio accuracy

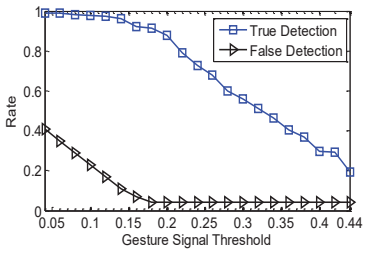
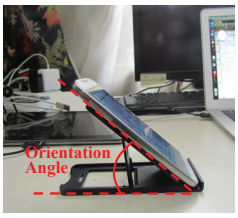


Figure 19: The gesture detection accuracy with gesture signal threshold



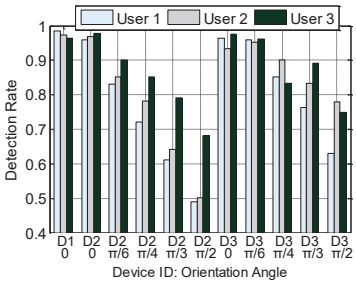


Figure 20: The device orientation angle with its detection accuracy

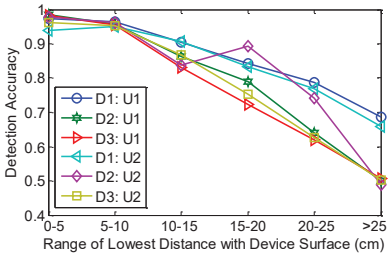


Figure 21: The device-hand distance with its detection accuracy

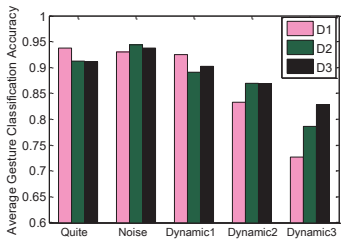


Figure 22: The average detection accuracy for different scenarios

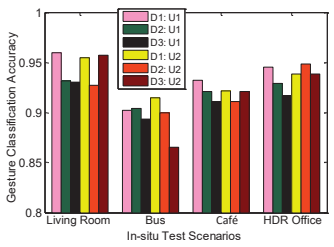


Figure 23: The average gesture classification accuracy for in-suit test

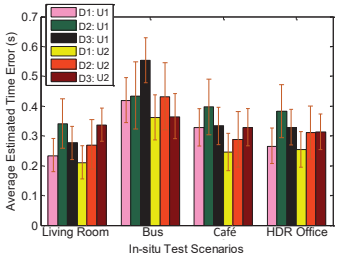


Figure 24: The average estimation error of hand in-air duration for in-suit test

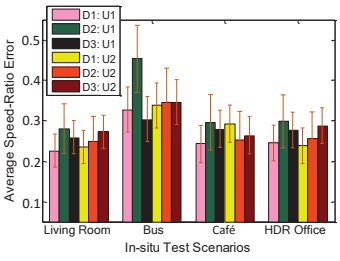


Figure 25: The average speed-ratio estimation error of hand movement for in-suit test

Table 1: Comparison of typical device-free localization systems

Comparison Items	WiGest[4]	FineGesture[18]	AllSee[15]	SoundWave[13]	SlideSwipe[36]	RadarGesture[20]	WiSee[25]	AudioGest
Measured Signal	RSSI	RSS, Phase, CSI	RF signal	Audio	GSM signal	FMCW Rada	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 track)	94%	95.1%

also decreases, which is mainly caused by the narrow space and unpredictable motion influences on the bus.

Discussion & Comparison

This section will briefly review our work and discuss the limitations that are left for future work.

Limited Hand Gesture Numbers: AudioGest can provide up to 162 control commands for applications by combining the hand-gesture types, hand in-air duration, average speed and waving range. It, however, can only distinguish eight hand gestures accurately. The main reason lies in that we only utilize one microphone and depend on the Doppler frequency shift to interpret the echo audio signal. In the future, we can either *i)* mine other features from the spectrogram of reflected signal to facilitate our physical model for recognizing more hand gestures (it may bring some burden of labeling training data); and *ii)* adopt 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 ($< 0.5m$). 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.

Comparing with the State-of-the-Art: Table 1 compares our AudioGest with other state-of-the-art gesture recognition systems. AudioGest thoroughly exploits the Doppler frequency shift from hand movement and further accurately interprets the spectrogram of echo signal into the hand gesture, in-air duration, hand average waving speed and moving range. AudioGest only uses one pair of COTS speaker & microphone without any extra hardware, and it is capable of sensing fine-grained gesture-contexts, *i.e.,* hand in-air duration, waving speed and range. It is training-free, and can provide up to 162 gesture control commands for various applications.

CONCLUSION

To summarize, this paper has shown how one single pair of microphone and speaker can real-time track human hand's radical velocity, thus decode the hand moving direction, estimate its waving speed and range. The real-world experiments demonstrate the feasibility and effectiveness of our system, which marks an important step toward enabling accurate and ubiquitous gesture recognition.

REFERENCES

1. Leap Motion, Inc. Leap Motion: Mac PC Gesture Controller for Game, Design and More. https://www.leapmotion.com/, 2013.
2. Nintendo. Wii console. http://www.nintendo.com/wii.
3. RoboRealm. Microsoft Kinect, http://www.roborealm.com/help/Microsoft Kinect.php, 2013.
4. Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. 2015. Wigest: A ubiquitous wifi-based gesture recognition system. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 1472–1480.
5. Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand. 2015. Capturing the human figure through a wall. *ACM Transactions on Graphics (TOG)* 34, 6 (2015), 219.
6. Fadel Adib and Dina Katabi. 2013. See Through Walls with WiFi!. In *Proceedings of the ACM SIGCOMM 2013 Conference (SIGCOMM '13)*. 75–86.
7. Sandip Agrawal, Ionut Constandache, Shravan Gaonkar, Romit Roy Choudhury, Kevin Caves, and Frank DeRuyter. 2011. Using mobile phones to write in air. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*. ACM, 15–28.
8. Parvin Asadzadeh, Lars Kulik, and Egemen Tanin. 2012. Gesture recognition using RFID technology. *Personal and Ubiquitous Computing* 16, 3 (2012), 225–234.
9. Gabe Cohn, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. Humantenna: using the body as an antenna for real-time whole-body interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1901–1910.
10. Nasser H Dardas and Nicolas D Georganas. 2011. Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. *Instrumentation and Measurement, IEEE Transactions on* 60, 11 (2011), 3592–3607.
11. G Deng and LW Cahill. 1993. An adaptive Gaussian filter for noise reduction and edge detection. In *Nuclear Science Symposium and Medical Imaging Conference, 1993., 1993 IEEE Conference Record.* IEEE, 1615–1619.
12. Han Ding, Longfei Shangguan, Zheng Yang, Jinsong Han, and others. 2015. FEMO: A Platform for Free-weight Exercise Monitoring with RFIDs. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems (SenSys '15)*. 141–154.
13. Sidhant Gupta, Daniel Morris, Shwetak Patel, and Desney Tan. 2012. Soundwave: using the doppler effect to sense gestures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1911–1914.
14. Kaustubh Kalgaonkar and Bhiksha Raj. 2009. One-handed gesture recognition using ultrasonic Doppler sonar. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 1889–1892.
15. Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota. 2014. Bringing gesture recognition to all devices. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. 303–316.
16. Hamed Ketabdar, Peyman Moghadam, Babak Naderi, and Mehran Roshandel. 2012. Magnetic signatures in air for mobile devices. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services companion*. ACM, 185–188.
17. David Kim, Otmar Hilliges, Shahram Izadi, Alex D Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 167–176.
18. Pedro Melgarejo, Xinyu Zhang, Parameswaran Ramanathan, and David Chu. 2014. Leveraging directional antenna capabilities for fine-grained gesture recognition. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 541–551.
19. Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Kari Pulli. 2015a. Multi-sensor system for driver's hand-gesture recognition. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, Vol. 1. IEEE, 1–8.
20. Pavlo Molchanov, Shalini Gupta, Kihwan Kim, and Kari Pulli. 2015b. Short-range FMCW monopulse radar for hand-gesture sensing. In *Radar Conference (RadarCon), 2015 IEEE*. IEEE, 1491–1496.
21. May Moussa and Moustafa Youssef. 2009. Smart devices for smart environments: Device-free passive detection in real environments. In *Pervasive Computing and Communications, 2009. PerCom 2009. IEEE International Conference on*. IEEE, 1–6.
22. Rajalakshmi Nandakumar, Shyamnath Gollakota, and Nathaniel Watson. 2015. Contactless sleep apnea detection on smartphones. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 45–57.

23. Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. 1515–1525.
24. Taiwoo Park, Jinwon Lee, Inseok Hwang, Chungkuk Yoo, Lama Nachman, and Junehwa Song. 2011. E-gesture: a collaborative architecture for energy-efficient gesture recognition with hand-worn sensor and mobile devices. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*. ACM, 260–273.
25. Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. 2013. Whole-home gesture recognition using wireless signals. In *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM, 27–38.
26. Siddharth S Rautaray and Anupam Agrawal. 2015. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review* 43, 1 (2015), 1–54.
27. W. Ruan. 2016. Unobtrusive human localization and activity recognition for supporting independent living of the elderly. In *Proceedings of 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*. 1–3.
28. Wenjie Ruan, Lina Yao, Quan Z. Sheng, and others. 2015. TagFall: Towards Unobstructive Fine-Grained Fall Detection based on UHF Passive RFID Tags. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS '15)*. 140–149.
29. Thad Starner and Alex Pentland. 1997. Real-time american sign language recognition from video using hidden markov models. In *Motion-Based Recognition*. Springer, 227–243.
30. Stephen P Tarzia, Robert P Dick, Peter A Dinda, and Gokhan Memik. 2009. Sonar-based measurement of user presence and attention. In *Proceedings of the 11th international conference on Ubiquitous computing*. ACM, 89–92.
31. Juan Pablo Wachs, Mathias Kölsch, Helman Stern, and Yael Edan. 2011. Vision-based Hand-gesture Applications. *Commun. ACM* 54, 2 (Feb. 2011), 60–71. DOI:**http://dx.doi.org/10.1145/1897816.1897838**
32. Sy Bor Wang, Ariadna Quattoni, Louis-Philippe Morency, David Demirdjian, and Trevor Darrell. 2006. Hidden conditional random fields for gesture recognition. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, Vol. 2. IEEE, 1521–1527.
33. Jiahui Wu, Gang Pan, Daqing Zhang, Guande Qi, and Shijian Li. 2009. Gesture recognition with a 3-d accelerometer. In *Ubiquitous intelligence and computing*. Springer, 25–38.
34. Lina Yao, Quan Z. Sheng, Wenjie Ruan, Tao Gu, Xue Li, Nick Falkner, and Zhi Yang. 2015a. RF-Care: Device-Free Posture Recognition for Elderly People Using A Passive RFID Tag Array. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS '15)*. 120–129.
35. L. Yao, Q. Z. Sheng, W. Ruan, X. Li, S. Wang, and Z. Yang. 2015b. Unobtrusive Posture Recognition via Online Learning of Multi-dimensional RFID Received Signal Strength. In *Proceedings of IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS'15)*. 116–123.
36. Chen Zhao, Ke-Yu Chen, Md Tanvir Islam Aumi, Shwetak Patel, and Matthew S Reynolds. 2014. SideSwipe: detecting in-air gestures around mobile devices using actual GSM signal. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 527–534.