IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 6, 2020, accepted February 1, 2020, date of publication February 11, 2020, date of current version February 20, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2973305

HCI on the Table: Robust Gesture Recognition Using Acoustic Sensing in Your Hand

GAN LUO[1,2], PANLONG YANG[1], MINGSHI CHEN[2], AND PING LI[2]
[1]School of Computer Science and Technology, University of Science and Technology of China, Hefei 230022, China
[2]School of Communications Engineering, The Army Engineering University of PLA, Nanjing 210001, China

Corresponding author: Panlong Yang (plyang@ustc.edu.cn)

This work was supported in part by the NSF of Jiangsu For Distinguished Young Scientist under Grant BK20150030, in part by the National Natural Science Foundation of China (NSFC) under Grant 61772546, Grant 61632010, Grant 61232018, and Grant 61371118, in part by the China National Funds for Distinguished Young Scientists under Grant 61625205, in part by the Key Research Program of Frontier Sciences, CAS, under Grant QYZDYSSWJSC002, Grant 61402009, Grant 61672038, and Grant 61520106007, and in part by the NSF under Grant ECCS-1247944, Grant NSF CMMI 1436786, and Grant NSF CNS 1526638.

ABSTRACT The paper proposes a new HCI mechanism for device-free gesture recognition on the table using acoustic signal, which can extend the gesture input and interactions beyond the tiny screen of mobile device and allow users to provide input without blocking screen view. Previous researches have either relied on additional devices (*e.g.*, special wearable device and mouse) or required active acoustic signals which demand additional cost and be less prone to popularize, while we explore the device-free gesture recognition using passive acoustic signals. This technology is more challenging due to the lack of an effective approach to eliminate the inherent ambient noise disturbances and extract stable gesture features. We fuse both short time energy (STE) and zero-crossing rate (ZCR) to identify the effective signals from the original input, and leverage the Mel frequency cepstral coefficients (MFCC), cochlear filter cepstral coefficients (CFCC) to extract the stable features from different gestures. The unique features in support vector machine (SVM) classifier achieve a high gesture recognition accuracy from the noisy scenarios and mismatched conditions. Implementation on the Android system has realized real-time processing of the feature extraction and gesture recognition. Extensive evaluations show our algorithm has a better noise tolerant performance and the system could recognize seven common gestures (click, flip left/right, scroll up/down, zoom in/out) on smart devices with an accuracy of 93.2%.

INDEX TERMS Human-computer interaction, acoustic sensing, gesture recognition, MFCC, CFCC, Android system.

I. INTRODUCTION

Smart devices such as smartphones and smartwatches have become pervasive and play a pivotal role in our daily life. However, the small size of the touch screen limits the user's experience and leads to many errors when user interacts with smart devices via on-screen soft keyboards. Besides, due to the small operational space on the touch screen, many Human-Computer Interaction (HCI) applications based on smartphones and smartwatches are becoming cumbersome and inconvenient. As a result, gesture recognition becomes a promising method for human-computer interaction and draws a lot of attention from research community.

One popular gesture recognition solution is to use computer vision (CV) techniques [1]–[3]. However, these CV based gesture recognition systems are generally sensitive to the ambient lighting conditions, and their implementations are usually too complex to be adopted on commercial smart devices. Another gesture recognition solution depends on specialized sensor hardware. To apply these methods, users have to either attach some delicate equipment on smart devices [4]–[6], or equip some wearable hardwares [7]–[9], which are burdensome, costly and inconvenient. For example, Soli [4] is a promising technique, however, it is expensive. To use Fingerpad [7], the user needs to wear fingertips which are inconvenient and easy to lose. Kinect [1] and Wii [2] can only identify coarse gestures and users are difficult to take them along. Fortunately, nowadays, smartphones have equipped many sensors and have more powerful computing

The associate editor coordinating the review of this manuscript and approving it for publication was Syed Anas Imtiaz.

capabilities than before, which make device-free gesture recognition possible. As a regular embedded sensor, the microphone is exploited to be used for fine-grained applications such as moving objects locating and tracking around the smartphone [10]. The similar technology that extends acoustic signal to gesture recognition has brought forth the prosperity of HCI applications.

Depending on different acoustic sources, the acoustic signal based systems can be divided into two prevailing modes, the active mode and passive mode. For active mode, a customized ultrasound signal is sent for motion sensing and object tracking such as SoundWave [11], FingerIO [12], Strata [13], UltraGesture [14], Vskin [15] and AcouDigits [16]. These methods can provide accurate motion sensing and tracking, however, they not only depend on extra sensors (*e.g.*, loudspeaker), but also consume more energy. For passive mode, it only leverages the embedded microphone to collect the sound caused by the moving objects, which makes them more energy efficient and more suitable for long-term sensing. For example, by fingers stroking on the keyboard, the emitted sound could be identified and recognized as different keystrokes [17] and [18]. SoundWrite [19] also uses the collected acoustic signal to recognize the user's input gesture. UbiWriter [20] leverages small mobile devices for recognizing freestyle handwriting via acoustic sensing.

In this article, we focus on the passive device-free gesture recognition for its energy efficiency and low complexity. In our scenario, the user does not need to wear any equipment, but only needs to draw different gestures on the plane near the smart device. Besides, the device can judge user's input gesture by analyzing the collected acoustic signal. To realize such system, three intrinsic challenges must be formally addressed.

- The first challenge is how to distinguish the acoustic signal corresponding to user's input gesture from the ambient noise. Obviously, the ambient noise will interfere the effective acoustic signal identification and affect the system performance consequently.
- The second challenge is how to choose the suitable acoustic feature. The selected acoustic feature should be conspicuously different among different gestures, and for the same gesture, the selected feature should be stable in various environments.
- The third challenge is how to select the appropriate classifier. As a human-computer interaction method, the system needs to recognize user input gestures quickly and accurately. Therefore, we need to make a trade-off between accuracy and real-time.

To solve the above challenges, we first design a dual-threshold scheme to identify the acoustic signal of the user's input gesture and separate it from the ambient noise. Compared with previous methods which are based on energy detection and preset threshold [19], [21], our dual-threshold segmentation scheme can identify and extract the acoustic signal corresponding to the user's gesture more effectively, and reduce the interference of ambient noise. Then, we fuse the Mel frequency cepstrum coefficient (MFCC) and cochlear filter cepstrum coefficient (CFCC) to address the second challenge better. Compared with amplitude spectrum density (ASD), MFCC and CFCC can provide acoustic characteristics in more detail. The MFCC works well in the scenario with high SNR while CFCC works well in the scenario with low SNR [22]. Combining them can provide more accurate and robust gesture recognition. Finally, we compared KNN and SVM classifier in terms of accuracy and time cost, and choose the SVM as our classifier for gesture recognition.

The contributions of our work are as follows:

- We present a dual-threshold signal segmentation scheme, which is more robust against to the ambient noise and can extract the useful signals effectively.
- We fuse MFCC and CFCC as a new acoustic feature. Comparing with ASD, our fused feature can provide better performance on both accuracy and robustness. In addition, we compare several popular classifiers and choose SVM as our classifier for its low time cost and acceptable accuracy.
- We implement the system and design the app on smartphone. Comprehensive experiments in real world show that our system can achieve a recognition accuracy of 93.2%, and has good robustness to ambient noise. Its performance is stable in different environments as well.

The rest parts of this paper are as follows: In Section II, a comprehensive introduction with technical details is presented for system design and show how to implement it on Android system in Section III. Then, we validate our design with experimental results and analysis in Section IV. After that, we discuss some related issues and review the related work in Section V and Section VI respectively. Finally, we conclude the paper in Section VII.

II. SYSTEM DESIGN

A. SYSTEM OVERVIEW

In this section we show an overview of the architecture and basic workflow of our system. Figure 1 gives a detailed and vivid demonstration of our gesture recognition system, which consists of four modules.

1) SAMPLING

Specifically, when users slide their fingers on the desktop, such as zoom in with thumb and forefinger, the friction between fingers and desktop will generate slight vibration. The microphone embedded in the smartphone will capture the acoustic signal caused by the vibration and acquire the original sampling of the acoustic signal.

2) EFFECTIVE SIGNAL SEGMENTATION

Then, our algorithm first utilizes the moving average window to eliminate background noise spectrum and interference, and the system will extract the effective parts of gestures from

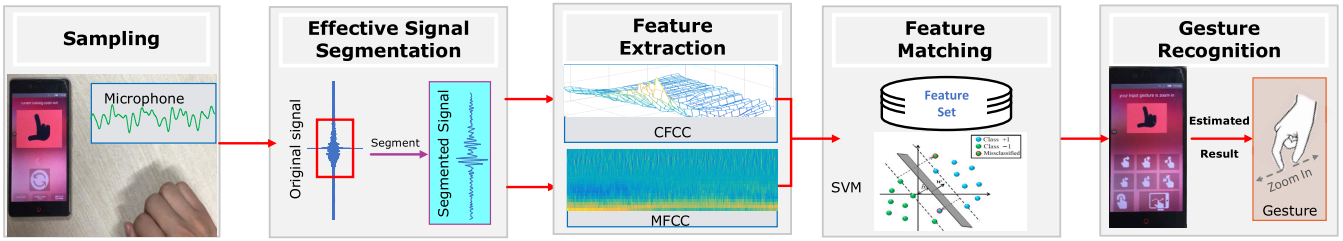


FIGURE 1. Algorithm diagram and system architecture.

such acoustic signal via a series of judgments and adaptive segmentation algorithm.

3) FEATURE EXTRACTION

In the next step, the unique feature of acoustic signal will be extracted and established for different gestures. Here we adopt the most representative acoustic feature named MFCC, and a new auditory-based feature named CFCC. Both acoustic features extraction algorithms imitate the human auditory system, and they have been proved as effective methods and have shown strong robustness in complex environment.

4) FEATURE MATCHING

Finally, we apply the machine learning algorithm (*i.e.* support vector machine (SVM) [23]) to estimate the input gesture. We first train each gesture and establish the feature set, and then leverage the SVM algorithm to match the extracted features of the input gesture with feature set and choose the optimal matching as the recognition result.

B. SAMPLING PROCESS

We record the acoustic signals of the gesture sliding on the table in a clean environment and extract amplitude spectrum density (ASD) by fast Fourier transform (FFT), as illustrated in Figure 2. From this figure we can see vividly that the frequency of gesture sliding is mostly below 2KHz, and the energy of the gesture signal is mainly concentrated in the low frequency spectrum (*e.g.*, less than 1KHz). According to the Nyquist-Shannon sampling theorem, if we want to get the complete gesture signal, the sampling frequency should be larger than 4KHz. On the other hand, the default sampling frequency of most mobile phones is 44.1KHz. The high frequency sampling frequency will increase the processing overhead, which may slacken the gesture recognition speed. Besides, high frequency signals will also involve more noise which affects the gesture recognition. Therefore, to avoid the negative impacts of high frequency sampling without reducing the sampling quality, we make a down-sampling (8KHz) before segmenting acoustic gesture signal.

C. EFFECTIVE SIGNAL SEGMENTATION

Ambient noise can interfere with the acoustic gesture sound and contaminate the signal spectrum. As a consequence,

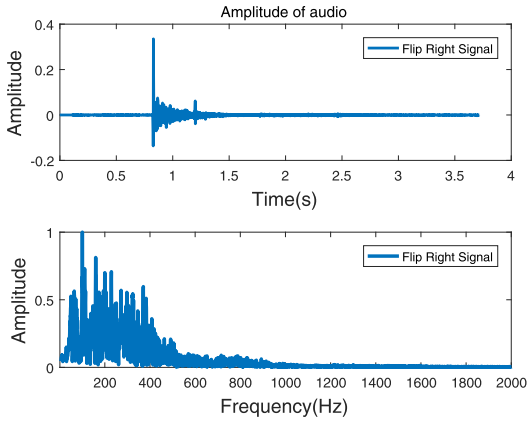


FIGURE 2. Amplitude spectrum density of acoustic signal.

extracting effective gesture segment from the raw recorded signal is the fundamental task for gesture recognition. Our acoustic signal segmentation module consists of three steps: filtering, denoising and segmentation.

1) FILTERING

As depicted in Figure 2, the effective gesture signal is concentrated below 2KHz, so we design a low-pass Butterworth filter with the cut-off frequency of 2KHz to remove the high-frequency component and obtain a clean gesture acoustic signal.

2) DENOISING

The inherent ambient noise strongly influences the performances of gesture identification systems. Therefore, to avoid the appearance of some burrs triggered by ambient noise in the acoustic signal, we use the moving average filtering to eliminate the interference of ambient noise. For each piece of acoustic signal $x(n)$, $A(t)$ denotes the moving average energy level with a window size W_a:

$$A(t) = \frac{1}{Wa} \sum_{n=t}^{t+Wa} x(n) \tag{1}$$

We set an empirical value $W_a = 5$ms, i.e., $8KHz \times 5ms = 40$ samples. After moving average filtering, the noise of the original acoustic signals will be reduced and eliminated, while the effective gesture signals will be enhanced and emerged.

3) SEGMENTATION

Prior works on gesture signal segmentation only relied on the energy feature [19], [24]. However, when the ambient noise is strong, applying the single energy feature is not suitable for gesture distinguish since the gesture feature is submerged by the noise. We present a dual-threshold scheme to extract the gesture signal from the raw signal. In this scheme, we combine zero-crossing rate (ZCR) [25] and short time energy (STE) together for signal segmentation. The new scheme is fast and has a relatively high accuracy for the remarkable discriminating ability of STE and ZCR [26].

The short time energy of the signal $x(n)$ is defined as

$$E_n = \sum_{m\to-\infty}^{\infty} [x(m) \cdot w(n-m)]^2 \tag{2}$$

where $w(n-m)$ is the window function used to extract a frame from the acoustic waveform.

The zero-crossing rate indicates the number of times a frame of an acoustic signal waveform crosses the horizontal axis (zero level). The average zero-crossing rate of a signal is defined as

$$Zn = \frac{1}{2} \sum_{m=-\infty}^{\infty} |sgn[x(m)] - sgn[x(m-1)]| \cdot w(n-m) \tag{3}$$

where sgn is symbolic function which is denoted as

$$sgn[x(n)] = \begin{cases} 1, & x(n) \geq 0 \\ -1, & x(n) < 0 \end{cases} \tag{4}$$

We divide the acoustic signal into multiple frames. Each frame contains 256 samples, the corresponding time is 32ms (256/8000HZ $\times$ 1000ms = 32ms). The duration of acoustic signal can be assumed as a stationary state [27], and the overlap between two adjacent frames is 80, which is about one-third of the frame length. Figure 3 shows the details of short time energy and the zero-crossing rate of gesture zoom in.

The judgment of the start and end points of the gesture signals is based on the following principles: the onset of gesture acoustic is indicated by a sudden large increase in acoustic energy as compared to the background energy level, while the termination of gesture acoustic is indicated by a smaller fall in energy level which may not be apparent in noise. We first normalize the amplitude of acoustic signal and set two thresholds for the short time energy. The initial low threshold of the short time energy T_L is 2, the high threshold T_H is 10. Similarly, we set the initial zero-crossing rate ZCR to 5. The choice of thresholds is based on the ambient noise level, and the thresholds of short time energy will make adaptive adjustment as algorithm 1 depicts. Then we set parameter *"maxsilence"* as the maximum mute length allowed in the gesture signal and *"minlen"* as the shortest length of the effective gesture signal segment. The initial *"maxsilence"* is set to 30 while the *"minlen"* is set to 20.

Algorithm 1 describes the details of the segmentation process. We divide the whole gesture signal into four states,

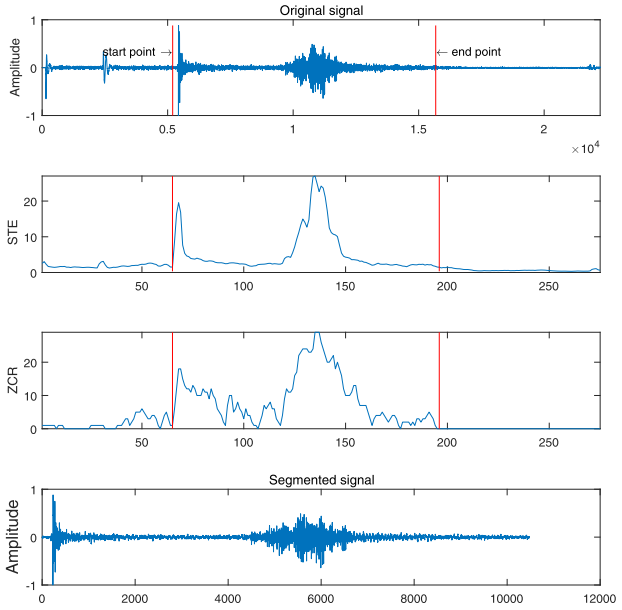


FIGURE 3. The STE and ZCR of gesture zoom in signal.

the state 0 represents the mute state, the state 1 represents a possible gesture state, the state 2 represents a gesture state, and the state 3 represents end state.

- In state 0 or state 1, for each frame of the acoustic signal $x(n)$, when the short time energy $E(n) > T_L$ or the zero-crossing rate $Z(n) > ZCR$, the frame signal is considered as the starting point of the signal and the parameter *count* is increased by 1, otherwise, the frame of acoustic segment will be regarded as noise and the algorithm will cut it off, the frame of acoustic signal will return to state 0. If $E(n) > T_H$, we confirm the frame is effective gesture signal and the parameter *count* starts to increase by 1, the acoustic signal will switch to the state 2.
- In the state 2, when $E(n) > T_L$ or $Z(n) > ZCR$, the parameter *count* is increased by 1, if the condition is not satisfied, then *silence* starts to increase by 1, if the length of *silence* is greater than *maxsilence* and the size of *count* is greater than *minlen*, we regard such frame as the end edge of the whole gesture signal, the acoustic signal will switch to state 3 and end the algorithm, otherwise the frame of acoustic signal will be considered as noise and re-judgment.
- In the state 3, the algorithm will break the loop and segment gesture signal, Figure 3 shows the final effective signal.

D. FEATURE EXTRACTION

How to select the most efficient and accurate feature is essential for gesture classification. In this module, we will describe the features we have adopted and compare the performance among different features. Finally, we will give the selected feature vector in our algorithm.

Algorithm 1 Algorithm for Segmentation

Input: The acoustic signal $x(n)$, and three temporal threshold $T_L = 2$, $T_H = 10$, $ZCR = 5$.
Output: The segment of each gesture $y(n)$.

Initialization: Status $= 0$, framelen $= 256$, framelnc $=$ 80, Count $= 0$, Silence $= 0$, $x_1 = 0$, $x_2 = 0$
Divide x(t) into frames $x_i(n)$;
for i_{th} *frame* $x_i(n)$ **do**
  Calculate $E(n)$ and $Z(n)$;
  $T_H = \min(T_H, \max(E(n)/4))$;
  $T_L = \min(T_L, \max(E(n)/8))$;
  switch Status;
  case 0 or 1
  if $E(n) > T_L$ *or* $Z(n) > ZCR$ **then**
    Status $= 1$; Count $=$ Count $+ 1$;
    if $E(n) > T_H$ **then**
      $x_1 = \max(i$ - Count-1,1);
      Status $= 2$; Silence $= 0$;
      Count $=$ Count $+ 1$;
    end
  end
  else
    Status $= 0$; Count $= 0$
  end
  case 2
  if $E(n) > T_L$ *or* $Z(n) > ZCR$ **then**
    Count $=$ Count $+ 1$;
  end
  else
    Silence $=$ Silence $+ 1$;
    if *Silence* > *Maxsilence* **then**
      if *Count* > *Minlen* **then**
        Status $= 3$; $x_2 = x_1 +$ Count-1;
        break the loop;
      end
      else
        Status $= 0$; Silence $= 0$;
        Count $= 0$;
      end
    end
    else
      Count $=$ Count $+ 1$;
    end
  end
end
$y(n) = x((x_1 - 1)\times$(framelen -framelnc) + 1:$x_2\times$(framelen -framelnc));

In our previous work, Soundwrite [19] has proposed a typical acoustic feature named amplitude spectrum density (ASD), which is the frequency domain profiles transformed from time domain signals by fast Fourier transform (FFT). Soundwrite extracts the unique ASD feature and calculates the position of the peak for each acoustic signal of the input gesture, since the ASD of each gesture exhibits distinct wave shape in the spectrum, and the peaks are different across frequencies, different gesture can be recognized and distinguished.

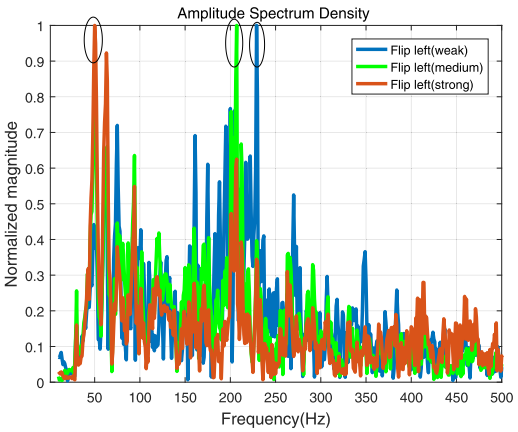


FIGURE 4. The ASD of one gesture in different strength.

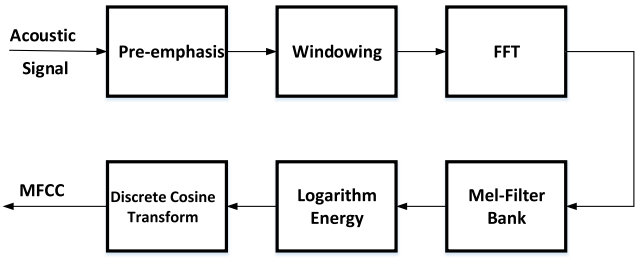


FIGURE 5. MFCC feature extraction diagram.

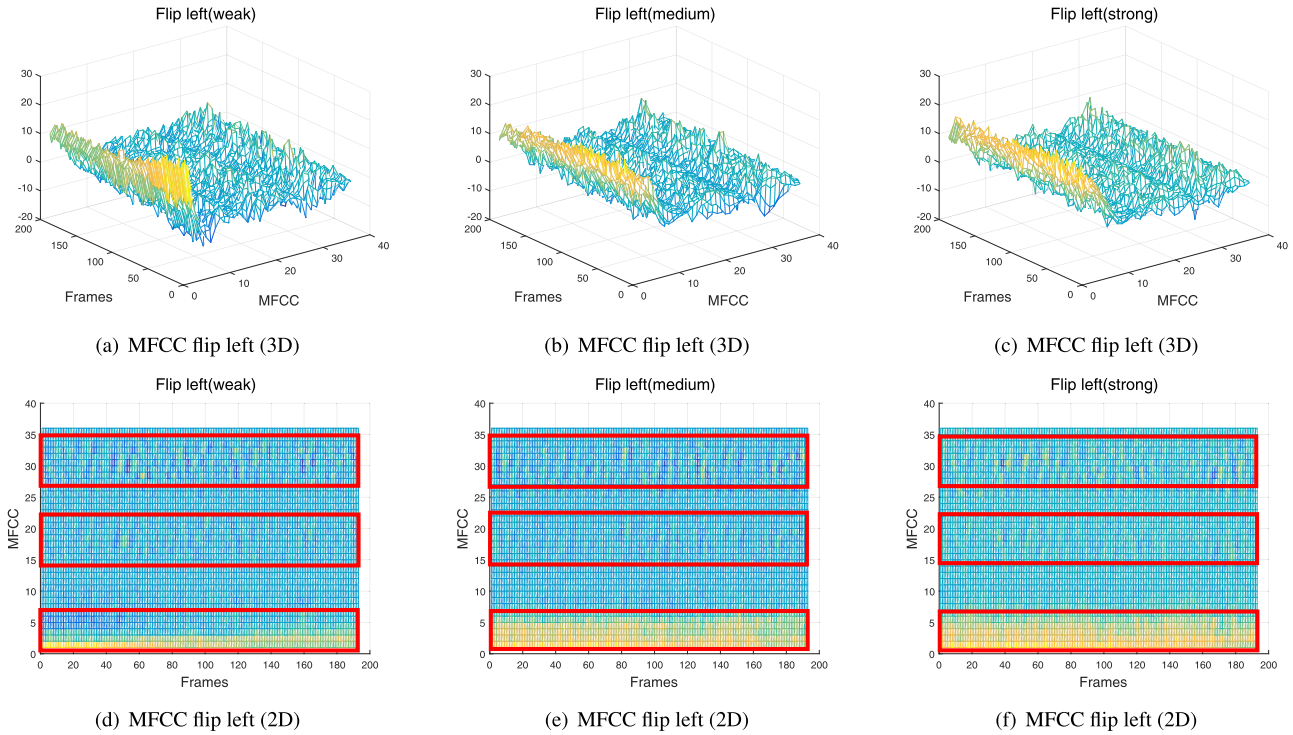
However, this method is not robust. When we input the same gesture on the table with different strength, the ASD will be different. Figure 4 shows the ASD of gesture flip left with different strength, we can see clearly that when we input gesture on the table with different strength, the peaks will have a conspicuous change. Besides, the ASD is the frequency domain characteristic of the whole input gesture signal, which is a global information, thus we cannot see the changes of acoustic signal over time.

1) MFCC FEATURE EXTRACTION

To avoid the deficiency of ASD, we adopt another feature named Mel frequency cepstral coefficients (MFCC) which is usually used in human speech recognition. MFCC takes human perception sensitivity with respect to frequencies into consideration. The mel-frequency cepstrum (MFC) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear Mel scale of frequency [28]. The extraction process of MFCC is shown in Figure 5.

The MFCC features are obtained by the below steps.

- Pre-emphasis: The process of pre-emphasis will make acoustic signal pass through a high-pass filter which can raise the high-frequency part and flat the spectrum of the signal.



(a) MFCC flip left (3D) (b) MFCC flip left (3D) (c) MFCC flip left (3D)

(d) MFCC flip left (2D) (e) MFCC flip left (2D) (f) MFCC flip left (2D)

FIGURE 6. The MFCC of one gesture in different strength.

- Windowing: We will multiply each frame by a hamming window to increase the continuity of the left and right ends of the frame. In our algorithm, the effective length of a frame is 256 sampling points.
- Fast Fourier transform (FFT): FFT is applied to each frame, to transform the distribution of energy into frequencies and calculate the periodogram of the power spectrum.
- Mel-filter bank: Apply the Mel filterbank to the power spectra, sum the energy in each filter.
- Logarithm energy: Take the logarithm of all filterbank energies. The transfer function that converts the linear spectrum $X(k)$ to the logarithmic spectrum $S(m)$ is:

$$S(m) = In\left(\sum_{k=0}^{N-1} |X(k)|^2 H_m(k)\right) 0 \le m \le M \quad (5)$$

- Discrete cosine transform (DCT): Take the DCT of the log filterbank energies and keep DCT coefficients.

$$C(n) = \sqrt{\frac{2}{M}} \sum_{m=1}^{M-1} s(m) cos\left(\frac{\pi n(m-\frac{1}{2})}{M}\right) \quad (6)$$

Finally, we extract the MFCC by following formula:

$$d_t = \begin{cases} c_{t+1} - c_t, & t < K \\ \dfrac{\sum_{k=1}^{K} k\,(c_{t+k} - c_{t-k})}{\sqrt{2\sum_{k=1}^{K} k^2}}, & others \\ c_t - c_{t-1}, & t \ge Q - K \end{cases} \quad (7)$$

where d_t denotes the t^{th} first order difference, c_t denotes the t^{th} cepstrum coefficient, Q denotes the order of the cepstrum coefficient, and K denotes the time difference of the first-order derivative, which is set at 1. The second-order difference vector can be obtained by taking the derivative of the first-order difference.

As mentioned above, we try to extract the information out of each gesture acoustic signal which consists of 50 samples and a total of 36 coefficients per signal. This feature extraction, though results in some loss of information, is sufficient for implementing classification techniques for basic feature detection.

We plot the MFCC of the gesture flip left in Figure 6. As Figure 6 shows, the profile of MFCC feature in the 3D view remains stable and the changes are not conspicuous. We convert the 3D versions of MFCC into 2D version for further processing. We mark the relatively obvious change areas of the MFCC with three red boxes in each sub-figure. From the trend of the MFCC changes in the figures and the variation range of the corresponding frame, we can see that the MFCC changes relatively smoothly, except the initial few frames, the other frames are relatively slight. Which means, compared with the ASD, the MFCC has certain robustness in distinguishing acoustic gestures signal.

2) CFCC FEATURE EXTRACTION

Although MFCC has better robustness in a clean testing condition [29], [30], when received strong ambient noise, the accuracy of the MFCC feature will decrease

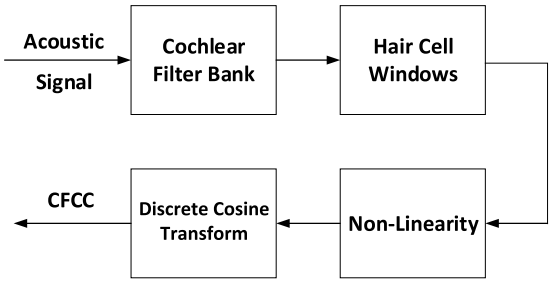


FIGURE 7. CFCC feature extraction diagram.

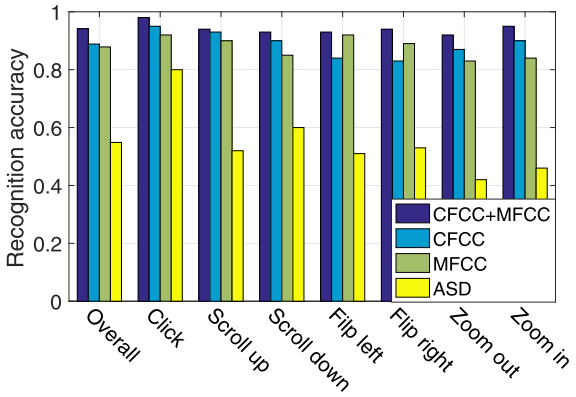


FIGURE 8. Average accuracy with CFCC,MFCC and ASD.

intensely [30], [31]. Considering the human hearing system is robust to the noisy conditions, we combine an auditory-based feature extraction algorithm which is modeled on the basic signal processing functions in the ear named cochlear filter cepstral coefficients (CFCC) [22].

CFCC is based on auditory transform (AT) and utilizes a set of modules to emulate the signal processing functions in the cochlea. The CFCC features have strong robustness in acoustic identification especially when the training and testing environments are mismatched [32]. The auditory feature extraction algorithm is shown in Figure 7.

The extraction process is mainly based on the human hearing system: the cochlear filter bank is intended to emulate the impulse response in the cochlea representing the basilar membrane (BM), the transformation of the filter models the whole process of the outer ear, middle ear and inner ear and uses the forward auditory transform to replace the fast Fourier transform used in many other features. The hair cell is a variable length window function, the nonlinear loudness transformation transforms energy information of the hair cells into perceived loudness. The discrete cosine transform (DCT) removes the correlation between the signals, the auditory features of acoustic signals through the above process are called cochlear feature cepstral coefficients.

The CFCC is extracted by following formula:

$$cfcc(n,j) = \sqrt{\frac{2}{M}} \sum_{m=1}^{M-1} y(m,j) \cos\left(\frac{\pi n(m-1/2)}{N}\right) \tag{8}$$

where $0 < n < N, 0 < m < M$, M is the number of cochlear filters, and N is the dimension of each frame feature after feature transformation. In this paper, $N = M = 18$.

MFCC feature is based on the Fourier transform (FT) which has a fixed time-frequency resolution and a well-defined inverse transform, while CFCC feature is auditory-based, time-frequency transform which is more similar to the mechanism in the human hearing system. Compared to the FFT, the AT has flexible time-frequency resolution and its frequency distribution can take on any linear or nonlinear form.

We compare different individuals and concatenative feature extraction techniques for system evaluation. Figure 8 is the recognition result of seven different gestures, we choose CFCC, MFCC and ASD as the feature for gesture recognition respectively, the figure further demonstrates that the combination of CFCC and MFCC has a significant improvement compared to ASD among all the seven gestures, and the gesture recognition rate has increased by about 30%. For most gestures (except gestures flip left/right) the CFCC features perform better than MFCC, while the fusion features of MFCC and CFCC perform the best.

Feature Vector: We first obtain the MFCC feature. Each sound segment is divided into frames with length Nc, and the overlap between two adjacent frames is N_f. For each frame, the number of cepstrum coefficients we adopt is N_m. So for each frame i $= 1, 2, 3 \ldots N$, we calculate a static vector of MFCCs: $[C_{m1}, C_{m2}, C_{m3} \ldots C_{mt}]$, and

$$C_{mi} = \frac{1}{N} \sum_{j=1}^{N} c_j^i, \quad i = 1, \ldots N_m \tag{9}$$

In our algorithm, the acoustic sampling rate is reset from 44.1KHz to 8KHz by down-sampling. We set $N_m = 36$, namely we choose 36 coefficients. The parameter N is positively correlated with the length of acoustic gesture segment effectively extracted by the algorithm, and c_j^i denotes the i^{th} MFCC values of j^{th} frame. The selection of parameter N_c and N_f depends on the specific experimental data. Generally speaking, the larger the parameter N_c, the more samples for each frame, and the worse the short-term stability of the acoustic signal. Conversely, the smaller the N_c is, the fewer samples for each frame is, and the better the short-term stability of the acoustic signal is. However, too few samples will increase the number of entire frames, which will increase the computational cost and degrade the real-time performance of the system. We set the samples N_c with different values for testing the recognition accuracy, and the overlap between two adjacent frames N_f is set to 1/3 of N_c. Figure 9 shows when N_c is set to 256 and the corresponding N_f is 80, the system recognition accuracy is the highest. Therefore, the algorithm set $N_c = 256$, $N_f = 80$. The CFCC feature extraction algorithm is similar to MFCC, for each frame, we calculate a static vector of CFCC: $[C_{c1}, C_{c2}, C_{c3} \ldots C_{ct}]$, and get the

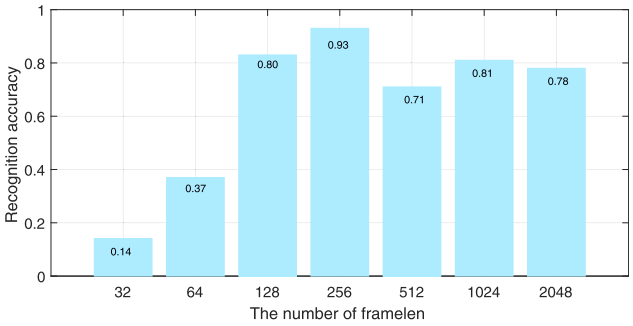


FIGURE 9. Average accuracy with different frame length.

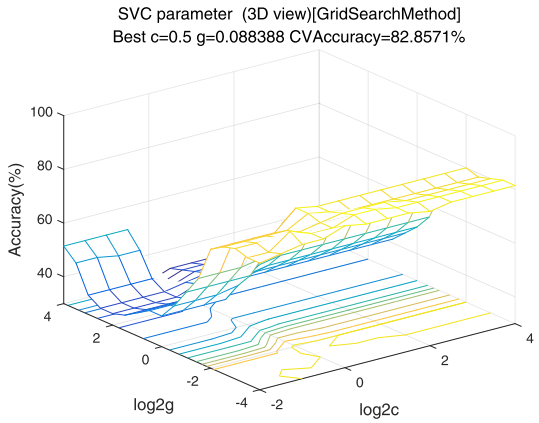


FIGURE 10. SVC parameter selection result (3D view).

mean value of each frame in the same dimension for entire acoustic signal.

For each frame, we extract the following features.

- MFCC: Which contains 12 MFCC values, 12 first-order difference parameters (ΔMFCC), and 12 second-order difference parameters (ΔΔMFCC).
- CFCC: which contains 18 coefficients.
- STE: Which shows the short time energy, the average of the STE denoted as

$$\bar{S} = \frac{1}{N}\sum_{1}^{N} E_n \tag{10}$$

  where E_n is the signal short time energy of each frame and N is the number of frames.
- ZCR: The zero-crossing rate, the average of the ZCR denoted as

$$\bar{Z} = \frac{1}{N}\sum_{1}^{N} Z_n \tag{11}$$

  where Z_n is the signal zero-crossing rate of each frame and N is the number of frames.

We denote the feature vector F for each acoustic signal fragment by:

$$F = [\alpha(C_{m1}, \cdots Cmt), (1-\alpha)(C_{c1}, \cdots Ccn), \bar{S}, \bar{Z}]$$

We combine MFCC features with CFCC features in a linear weighting method. The fusion coefficient α depends on the environments and will make adjustment according to the surrounding noise. In general, the stronger the ambient noise level, the lower the coefficient of α is.

E. FEATURE MATCHING

We used the support vector machine (SVM) supervised learning algorithm to tackle the classification and recognition of gestures. SVM constructs a hyperplane or a set of hyperplanes in a high dimensional space, which can be used for classification or regression. For input samples in n-dimensional space, it looks for an optimal classification hyperplane, so that two types of samples can get the best classification in this hyperplane. SVM is essentially a two-class classifier, however, it can be extended to a multi-class classifier. The common methods are one-versus-many discriminant strategy and one-versus-one discriminant strategy. For the former, in the training, it classifies one category or object as one class in the k-class samples, and the rest is classified into another class. So, the k-class samples need k two-classifier. When classifying unknown samples, the k two-classifiers are used for classification respectively, the category that appears most frequently in the classification result is the final classification. For the latter, any two types of samples will train a two-classifier, and a k-classifier is consist of $k*(k-1)/2$ two classifiers. When classifying unknown samples, all k*(k-1)/2 classifiers are used for classification, and the category which occurs the highest frequency is used as the final classification result of the sample. In our algorithm, we choose the one-versus-one discriminant strategy, and the kernel function we have chosen is radial basis function (RBF) kernel which is defined as:

$$k(x_1, x_2) = e^{(-\frac{||x_1 - x_2||^2}{2\sigma^2})}, \quad \sigma > 0 \tag{12}$$

The above equation can also be written as:

$$k(x, y) = e^{-\gamma||x-y||^2}, \quad \gamma > 0 \tag{13}$$

the reason we chose RBF is that it can achieve nonlinear mapping and has relative small number of parameters. When introducing the relaxation factor $\xi(i)$, the objective function and constraint condition for SVM are:

$$\begin{aligned} &min\ \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi(i) \\ &s.t.\ y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi(i), \\ &\quad \xi(i) \geq 0, \quad i = 1, 2 \ldots n \end{aligned} \tag{14}$$

where w and x are n-dimensional column vectors, b denotes the distance from the hyperplane to the origin and c denotes the penalty coefficient. In SVM, there are two important parameters c and g, c controls the overfitting of the model, and g (γ) controls the degree of nonlinearity of the model. The two parameters can get the optimal value by k-fold cross validation and grid-search as Figure 10 shows.

TABLE 1. Performance between SVM and KNN.

Method	Instances	Accuracy	Time(s)
SVM (c=0.5, g=0.08)	5	73.00%	0.34
	10	84.00%	0.50
	15	91.30%	0.66
	20	93.2%	0.78
KNN (k=7)	5	57.00%	0.09
	10	80.00%	0.16
	15	90.00%	0.29
	20	91.1%	0.35

As a comparison, we also used the k-nearest neighbors (KNN) algorithm for feature classification as it has a relatively simpler implementation on smartphone. For the sake of convincible demonstration on our algorithm, we train seven gestures and each gesture is repeated 20 times. With the increased parameter k, we conduct the gesture recognition experiments respectively and establish the training set. In our experiments, the optimal set of parameter k we have chosen is 7 from the statistical accuracy rate, and the corresponding recognition accuracy could be 91.4%. Table 1 presents the recognition accuracy and computation time for different training instances between SVM and KNN. We can see SVM has a better performance especially when the training instances is relatively small. Both SVM and KNN achieve high accuracy as the training instance increases, however, the growth rate of time cost in KNN is significantly higher than that of SVM.

III. SYSTEM IMPLEMENTATION

In this section, we introduce the technical details of our system implementation on Android mobile phones.

A. SAMPLING PROCESS

As for Android system design, we collect the acoustic signal by invoking the *"AudioRecorder"* function, which is embedded in android studio. The framework of sampling module is shown in Figure 11. The Android app contains three layers: user layer, audio driver layer and hardware layer. The user layer provides a lot of APIs for user to develop gesture recognition applications, we can implement the entire function of android audio at user layer, while audio driver layer and hardware layer cannot be modified by the user and their control permissions lie in the system. The user layer can be further decomposed into four layers in detail: app layer, framework layer, audio lib layer and HAL (hardware abstraction layer). At the APP layer, we state *"android.permission.RECORD_AUDIO"* and *"android.permission.WRITE_EXTERNAL_STORAGE"* to get the record and store permissions of the microphone. We collect audio gesture signal by invoking *"AudioTrack"* function of android studio and record the audio data by invoking another API named *"AudioRecorder"*. The audio Lib layer is an interface, it contains a series of library functions for

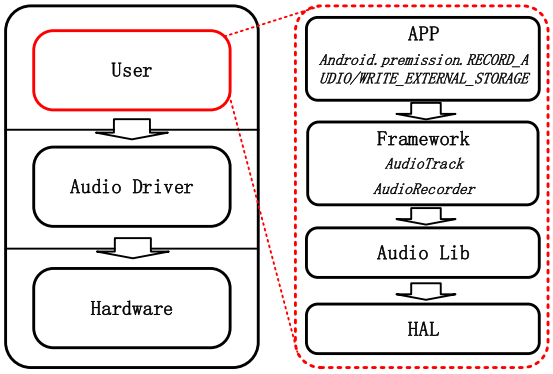


FIGURE 11. Framework of sampling module.

audio processing. The hardware abstraction layer is mainly responsible for managing audio equipment.

B. EFFECTIVE SIGNAL SEGMENTATION

As for android design, we first use the *"Average()"* function to calculate the moving average energy and eliminate the ambient noise. Then we invoke signal processing toolbox of Matlab in Android Studio. We utilize *"Enframe()"* function to frame the signal, next step, we use the *"energy()"* function and *"zerocross()"* function to calculate the short-term energy and zero-crossing rate, respectively. At last, we invoke *"System.arraycopy()"* function to find the split point of the gesture signal. Finally, we extract gesture segments through a series of key points.

C. FEATURE EXTRACTION

Now we describe the extraction and implementation of feature vectors of Android system. Taking MFCCs for example, in the pre-processing stage, the signal is subdivided into frames according to the effective acoustic gesture segmentation. Then we invoke the *"FFT.java"* class for fast Fourier transform, and convert the acoustic gesture signal of each frame from time domain into the energy information of the frequency domain. Next, *"calculateMelBasedFilterBank()"* function which is embedded in the *"MFCC.java"* class [33] will be invoked to obtain the Mel filter banks. In order to eliminate the effect of harmonics and highlight the resonant peak of the original acoustic signal, we smooth the signal through 20 sets of Mel-scale filter banks. The Mel filter protects the MFCC from the affection of the signal strength variation and reduces the computational complexity as well. After that, we use the *"Math.log"* function to calculate the logarithmic energy output for each filter bank. Finally, the above-mentioned logarithmic energy is multiplied by the DCT matrix calculated by the *"initializeDCTMatrix()"* function of the *"MFCC.java"* class, and the discrete cosine transform is performed to obtain the 12 MFCCs values. We get the static values from the *"getParameters()"* function of the *"MFCC.java"* class. Here, along with MFCC coefficients we have also used delta MFCC coefficients which represent the change in frequency power.

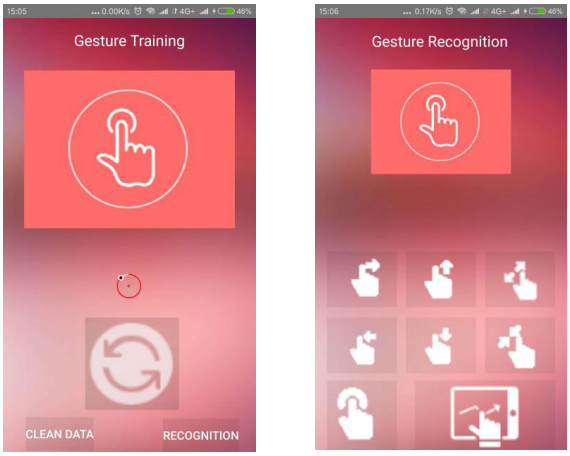


FIGURE 12. The interface design of the APP.

D. FEATURE MATCHING

Here we introduce the android implementation of feature matching module. In Android Studio we invoke the LIBSVM library [34], which is a simple and efficient software for SVM classification and regression. In order to find the best parameters c and g, we invoke *"SVMcgForClass()"* function. The *"svm_train()"* function is used to train the classification model while the *"svm_predict()"* function is used to predict the recognition result.

Finally, we briefly introduce our app, whose interface design mainly includes two modules: acoustic gestures training module and acoustic gestures testing module, as demonstrated in Figure 12. In the training stage, for each gesture the system will remind the user to input the gesture within 3 seconds, while in the testing stage the system will automatically detect the input gesture signal and display the recognized gesture. Meanwhile, the error recognition can also be corrected according to the user's judgment, and the rectificatory results will be updated to the training data set automatically.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed algorithm in terms of recognition accuracy and robustness by conducting a series of experiments in the real environment. We have conducted extensive evaluations on macro benchmark for system-level performance and the micro benchmark for component-based evaluation with various influencing factors.

A. EXPERIMENT SETUP

The experiments are conducted in laboratory, dormitory and classroom with different ambient noise levels. For each environment, 10 volunteers (7 males, 3 females) are invited to input gestures on different materials. The volunteers will input seven different gestures in the training stage. For each gesture, the duration of the training and test acoustic signals is set to 3 seconds. We will count the recognition accuracy of each gesture separately in the testing stage. We installed the app on three different Android smartphones (Samsung S5, HUAWEI Mate9, Xiaomi MI6). As the default setting, a Xiaomi MI6 smartphone is placed on a coated-wood table in the laboratory, and the audio sampling frequency is set to 44.1kHz in order to obtain high quality acoustic signal for gesture recognition. Figure 14 is a demonstration of our experiment scenario. The volunteer inputs the gesture on the table while the smartphone near the user collects the acoustic signal and recognizes the gesture.

B. MACRO-BENCHMARK

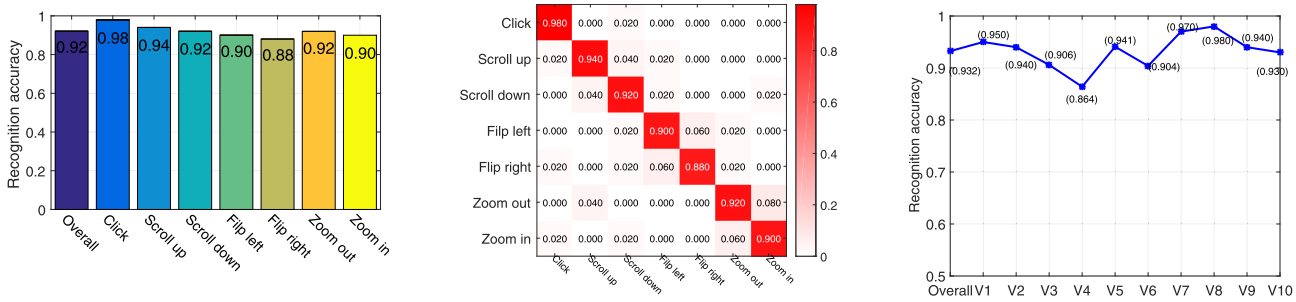
1) AVERAGE ACCURACY OF EACH GESTURE RECOGNITION:

We first evaluate the average recognition accuracy for each gesture. In this experiment, we have specified a list of gestures for each volunteer and required them to input gestures in the order on the list in a clean environment. In order to mimic the daily use, in each list, all gestures have the same input serial number, but the order is set randomly. Each gesture is scattered among the list, and both the training instances and testing instances of each gesture are 20. The volunteers firstly input each gesture 20 times to establish the training set. Then, in the testing stage, they randomly input each gesture and judge the recognition results. Finally, the volunteers mark the recognition results on the list. After the experiment, we collect all the lists and count the average accuracy of each gesture recognition. Figure 13(a) shows the average recognition results of 10 volunteers, where the average accuracy of the correct gesture recognition rate is about 92%. The gesture "click" has the highest recognition accuracy, which is 98%, while the gesture "flip right" has the lowest recognition accuracy at 88%.

To deeply investigate the recognition accuracy among all gestures, we count the estimation results of our algorithm on each gesture and present them in the form of a confusion matrix. As Figure 13(b) illustrates, estimation errors are more likely to occur between the same set of gestures in different directions. For example, the gesture "flip left" is prone to be estimated as "flip right", gesture "scroll up" and "scroll down" are likely to confuse with each other, and gesture "zoom out" tends to be estimated as "zoom in". The reason behind the phenomenon is that these confused gestures have a relatively high similarity except the direction, thus the corresponding acoustic features are hard to be distinguished.

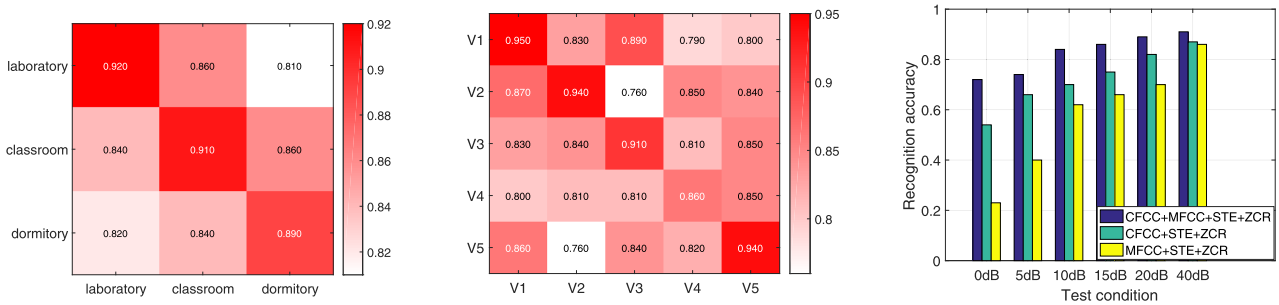
2) AVERAGE ACCURACY OF DIFFERENT USERS

We then test the average accuracy among different users. In this experiment, we leverage the data in the previous experiment and calculate the average gesture recognition accuracy of each user. Due to the difference in sliding habits of different users, the gesture signal will vary in spectrum, which may affect the accuracy of the corresponding gesture recognition. Figure 13(c) plots the average accuracy of 10 volunteers. The figure intuitively shows that although there is a difference in the recognition rate among different users, the recognition accuracy of each user is still with an average of 93.2%. It demonstrates the robustness of the unique features and the effectiveness of the algorithm.



(a) with different gesture (b) confusion matrix (c) with different users

FIGURE 13. Average accuracy evaluation over different users and gestures.



(a) with mismatched scenarios (b) with mismatched individual (c) with different SNR

FIGURE 15. Average accuracy evaluation over mismatched conditions.

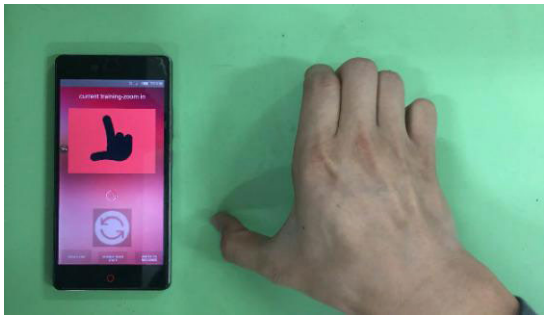


FIGURE 14. Experimental scenario.

3) AVERAGE ACCURACY WITH MISMATCHED SCENARIOS

In our previous experiments, the training data set and testing data set are conducted in the same scenario, and our algorithm maintains a high-level recognition accuracy. We would like to know whether our system can perform well even the training environment is not the using environment. We evaluated our system in a task where the acoustic conditions of training and testing are mismatched, *i.e.*, the training data were conducted under one environment while the test is conducted in another place. We choose three typical scenes (laboratory, classroom, dormitory) for testing, the noise level is about $35dB$,[1] $48dB$ and $55dB$, respectively. Figure 15(a) shows the average recognition accuracy with different training scenarios and testing scenarios. The X-axis is the training place while the Y-axis is the testing place. As shown in the figure, when the training scenarios and testing scenarios are the same, the accuracy is above 89% with 7 typical gestures. Even the training scenarios and testing scenarios are mismatched, the average recognition accuracy can still reach up to 81%. The rationale behind this is that, although the acoustic features generated by different scenarios are diverse, with each person corresponding to their training set, the accuracy of gesture recognition is impervious in other places. In other words, each user's input habits and gesture features are also included in the extraction feature of our gesture recognition algorithm.

[1]The dB (decibel) here is the sound intensity level

4) AVERAGE ACCURACY WITH MISMATCHED INDIVIDUAL:

Due to the diversity in input gesture habits, the feature established by different users are various. We evaluated the impact of the training set for mismatched individual. We reuse the data in Figure 13(c) and select the data of the top five volunteers. We choose the gesture data from one of the five volunteers as the training set, the other input gesture data as the testing set, for each gesture we observe the recognition accuracy of the mobile phone and count the identification results. As shown in Figure 15(c), although the training set and the testing set are not extracted from one person, our algorithm still maintains a relatively high-level recognition accuracy, the recognition accuracy could be 88% with

7 typical gestures. The explanation is that the input habits of different users only affect the features in the time domain of the acoustic signal, and the influence in the frequency domain is not obvious.

5) AVERAGE ACCURACY WITH DIFFERENT LEVELS OF SIGNAL PER NOISE RATIO (SNR)

The ambient noises (human voice, traffic noise, and life noise) have a non-negligible influence on gesture detection and recognition. We conducted gesture identification experiments on the testing set with different levels of SNR. The training acoustic signal of gesture was recorded under clean testing environment, and then different noise is added to the clean testing data at increasing intensity, there are six testing conditions in the experiment (*i.e.*, noisy level at $0dB$,[2] $5dB$, $10dB$, $15dB$, $20dB$ and $40dB$ SNR). From Figure 15(c) we can see that the SNR level directly affects the performance of the gesture recognition accuracy. When the SNR is reduced to $0dB$, the gesture recognition rate could achieve 72%. The performance of the algorithm will drop dramatically as the SNR decreases when we only use MFCC. When the SNR drops to $0dB$, the accuracy is only about 23%, while the accuracy can achieve 54% when we use CFCC. Neither of them can compare with the combination MFCC and CFCC, the gesture recognition rate could achieve 72%. When the SNR increase to $40dB$, both MFCC and CFCC can achieve high accuracy. This experiment proves that the CFCC has shown stronger robustness than MFCC in noisy environment, and the MFCC combined with the CFCC will get the best score in terms of accuracy.

C. IMPACT FACTORS

To have a deep understanding of our system, we conduct extensive experiments to evaluate the impact of some key factors on our algorithm performance.

1) AVERAGE ACCURACY IN DIFFERENT SCENARIOS:

To present the evaluation of our algorithm in real-world environments, we selected three typical scenes to test: 1) a laboratory with people moving around, the corresponding ambient noise level is $35dB$; 2) a classroom with some students talking, the corresponding ambient noise level is $47.8dB$; 3) a relatively noisy dormitory, the corresponding ambient level is $59.8dB$. The volunteers are required to input all 7 gestures and repeat each gesture 20 times in each scenario. We measure both the error rate and recognition accuracy of the gesture inputs to evaluate the performance of gesture recognition

As depicted in Table 2, the ambient noise has a significant negative impact on gesture recognition. In general, the noise intensity has a negative correlation with the accuracy of gesture recognition, that is to say, the stronger the noise, the lower the accuracy of gesture recognition. The mis-detection (P_{mis}, false negative) and false-alarm(P_{fls}, false positive) rates are used to measure the gesture input detection rate. We can see that when the noise level is lower than 60dB, the P_{mis} and P_{fls} is very low, namely, our system has a very high detection sensitivity and a very low misjudgment rate.

[2]The dB here is the unit of SNR

TABLE 2. Accuracy with Different Scenarios.

Environment	laboratory	classroom	dormitory
Nosie level	35dB	47.8dB	59.8dB
Accuracy	91.2%	90.62%	85.49%
P_{mis}	0.49%	1.69%	2.26%
P_{fls}	0.0%	0.60%	1.94%

TABLE 3. Accuracy with different materials.

Material	paper	cardboard	wood	metal
Accuracy	85.6%	89.5%	93.2%	91.3%

2) AVERAGE ACCURACY WITH DIFFERENT SURFACE MATERIALS

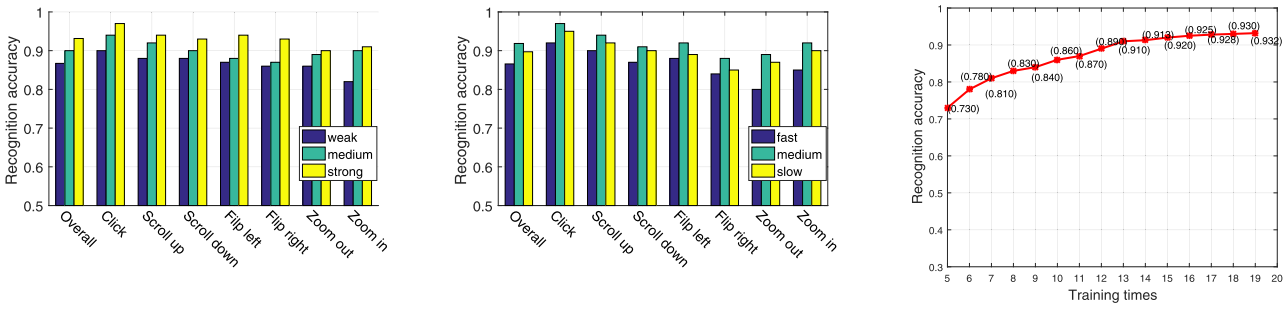
Different materials may have different effects on the gesture signal due to the difference in hardness and friction coefficient. In order to investigate the relationship among different surface materials and gesture recognition, we considered four common materials, paper, cardboard, wood and metal. In this experiment, we placed the phone on the above materials, respectively, and then volunteers input gestures among these materials in turn.

As Table 3 illustrates, different materials have a discrepant impact on the accuracy of gesture recognition. Specifically, the accuracy of metal and wood has a better performance than that on paper and cardboard. The reason is that the surfaces of metal and wood are rough and solid which emit larger amount of detectable acoustic signal, while the surface of paper and cardboard are smooth and soft which make the sound weak. The strong acoustic signal power contributes to eliminating ambient noise and extracting gesture features, and with more significant and sufficient information, the system will perform better and achieve high gesture recognition accuracy.

3) AVERAGE ACCURACY WITH DIFFERENT SMARTPHONES

Different smartphones have various hardware configurations and microphone. As a consequence, the acoustic capture ability of different smartphones may not be equal. In order to verify the stability of our algorithm on different smartphones, we select three different types of mobile phones (Xiaomi MI6, Samsung S5 and HUAWEI Mate9) to test and compare their recognition accuracy of different gestures. The experimental results are shown in Table 4.

As Table 4 shows, the distinction in hardware configuration makes the gesture recognition have a small variations among different smartphones, and the average accuracy performance is kept at around 90%. The accuracy performance on HUAWEI Mate 9 achieves the best performance, while



(a) with different strength (b) with different sliding speed (c) with different training instances

FIGURE 16. Average accuracy evaluation over different input behaviours.

TABLE 4. Accuracy with different smartphones.

Device	Xiaomi	Samsung	HUAWEI
Accuracy	91.32%	90.35%	93.16%

Xiaomi MI6 and Samsung S5 achieve slightly lower performance. This is because Mate 9 is equipped with four microphones which can capture the slighter sound produced by the finger sliding.

4) THE IMPACT OF TOUCH STRENGTH ON AVERAGE ACCURACY

One common concern about the impact factor for gesture recognition is the touch strength. Intuitively, the input strength may be at strong, medium and weak level. Usually, the intensity of gesture input varies from person to person because of different input preferences among volunteers. In the experiment, we required the volunteers to input gesture with three different touch strength, the intensity was 3.56N, 1.76N and 0.75N, respectively, and N is the unit of force. These are derived from the mechanical formula $F = mg$ in physics, where m is the mass and g is the gravitational acceleration, the minimum resolution of the input intensity in weighting scale is 0.01g. The statistical results are depicted in Figure 16(a), which clearly demonstrates that the gesture recognition accuracy is positively correlated with the sliding strength. When our fingers slide on the desktop, different touch strength will lead to different intensity of friction. Generally speaking, the weak friction signal is easily submerged in strong ambient noise, while strong friction sound is conducive to gesture signal extraction and feature matching.

5) THE IMPACT OF SLIDING SPEED ON AVERAGE ACCURACY

Another impact factor for gesture recognition we evaluate in the experiment is the sliding speed. The sliding speed of gesture directly affects the quality of signal acquisition and the length of the effective segmentation gesture signal. We invite the volunteers to slide their fingers on the table with different sliding speeds. For instance, the volunteers input gesture "flip left" at the sliding distance 10cm within 1.5s, 0.7s and 0.3s, respectively, the amplitude of gesture "flip left" with different sliding speed is shown in Figure 17.

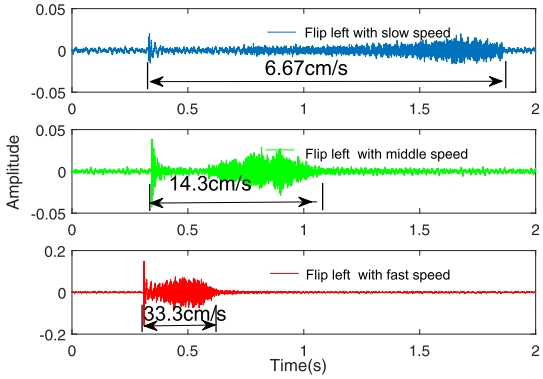


FIGURE 17. The amplitude of gesture with different sliding speed .

Figure 16(b) illustrates the collected results, from the figure, we can distinctly see that compared with the medium sliding speed, too fast or too slow gesture sliding will have a negative impact on the recognition accuracy of the system. This phenomenon is explained as follows, the number of sampling points for each divided frame in MFCC and CFCC is constant, the slower the slide speed, the larger amount of frame sequence will be extracted, which leads to more ambient noise to be mixed up with the effective gesture feature, thus obstructs the final feature matching. On the other hand, too fast slide speed will reduce the length of effective acoustic signal, namely, less frame sequence information can be utilized for the feature extraction, and the accuracy drops sharply consequently.

6) THE IMPACT OF TRAINING SET ON AVERAGE ACCURACY

Machine learning algorithms are generally highly dependent on the number of samples. Without a large number of training samples, there is no good training model. Figure 16(c) shows the recognition results of the SVM algorithms with the increase in the number of training set in detail. As the figure demonstrates, when the initial training set is five, SVM algorithm doesn't perform well, and with an extremely low level of recognition rate. However, when the number of sam-

ples escalates to 15, the system recognition rate has reached 91.3% in SVM. Marginal improvement is achieved by further increasing the number of training instances beyond 15.

V. DISCUSSION

A. FEATURE EXTRACTION TECHNIQUE

This section compares some mainstream feature extraction techniques such as fusion of feature warped MFCC and discrete wavelet transform (DWT), and the fusion of MFCC feature and other features such as power normalized cepstral coefficients (PNCC).

1) FEATURE WARPED MFCC & DWT

Strong noise and channel distortion will corrupt the logarithmic energy of MFCC and lead to nonlinear distortion of the distribution of the cepstral features over time [35]. Feature warped MFCC [36] was used to compensate this nonlinearity and generate a stronger distribution representation for each cepstrum feature. Besides, the discrete wavelet transform (DWT) is a prevalent tool for analyzing the acoustic signal in time and frequency. Many researchers have introduced the wavelet transform (WT) into the extraction of feature warped MFCC (FW-MFCC), for instance, Ahmed Kamil Hasan *et al.* combine FW-MFCC and DWT for speaker verification [37], [38]. They apply DWT to decompose the speech into the low frequency sub-band and the high frequency sub-band coefficients, and then splice the frequency response of the wavelet coefficients into a complete frequency spectrum, finally, they acquire the Mel logarithmic power spectrum by calculating the wavelet coefficient energy. The experiment shows the fusion of FW-MFCC and DWT will decrease the performance of speaker verification system. The reason behind this phenomenon is that the fusion lost some crucial correlation information between sub-band features, the lack of the correlation between these small bands brings negative influence on the speaker verification system [39]. Compared with this method, we use the original MFCC feature, it contains sufficient spectrum information to recognize the gesture signal.

2) MFCC & PNCC

The feature power normalized cepstral coefficients (PNCC) is based on auditory processing, and it shows higher robustness than MFCC in noisy environment [40]. Kim *et al.* explore the combination of MFCC and PNCC features for robust biometric speaker identification [41]. They use three fusion methods (fusion maximum, fusion mean and fusion weights) for the two acoustic features, the experiments demonstrate the combination of MFCC and PNCC can achieve higher identification rate than using each feature separately. However, the PNCC has a relatively higher error rate in the clean environment, because the power-bias subtraction will result in signal distortion when removing the background noise [40]–[42]. Compared with the above features, the CFCC performs consistently high accuracy in noise environment especially under a variety of mismatched testing conditions. Even in clean condition, the CFCC achieves a high accuracy and outperform the PNCC [22]. In comparison, the CFCC feature generates superior results under real environment, and the fusion of MFCC feature and CFCC feature has shown robustness in various experiments.

B. VOICE ACTIVITY DETECTION (VAD)

Nowadays, a host of researches for voice activity detection (VAD) have emerged. Apart from the method based on the short time energy and zero-crossing rate, another technique which is widely used is statistical model. The method derives from the likelihood ratio test (LRT). It assumes that the speech and background noise are independent distributions, and then calculates the model parameters of each frame of signal respectively, and finally detects the effective speech signal by calculating the likelihood ratio [43]. Though the approach based on statistical model achieves high detection recognition rate in speech recognition, it relies on the accuracy of the model, and the detectability on speech-like noise is poor. The methods often hypothesize the discrete Fourier transform (DFT) coefficients of noise is single Gaussian distribution for simplifying the calculation, although it is not the case in practice. This problem leads to the methods based on statistical model give a bad distinguishing ability in low SNR especially in non-stationary noisy conditions [44]. Besides, the complicated computation of the method increases the time cost and is not conducive to implementing on smartphone in real time. In order to model the noise or characterize the speech distribution, they often use large training sets to obtain Gaussian mixture models (GMMs) [45]. On the contrary, the method based on zero-crossing rate and short-time energy has low time cost and is suitable for real-time applications on mobile phone. The dual-thresholds methods can reach high recognition accuracy with low complexity, it is highly desirable for acoustic signal processing applications especially in real-time systems.

C. CLASSIFIER TECHNIQUE

Currently, the common classifier techniques for gesture recognition mainly include machine learning and deep learning. We will give a detailed discussion about these methods.

1) MACHINE LEARNING

Machine learning methods extract specific acoustic features as the input data to classify gestures, *e.g.*, SVM, hidden Markov model (HMM), length normalized GPLDA. HMM is a kind of statistical analysis model and can effectively handle time-varying sequences. Many researchers adopt HMM model to classify hand gesture and achieve satisfying performance. DuG [46] utilizes two speakers and a microphone to recognize a set of 11 gestures under a common HMM model, the system maintains average 98% accurate gesture recognition. Despite the high recognition, the HMM model needs a long time to train and the recognition time is too long [47], the complexity of HMM is high in the compu-

tational process [48]. While SVM has better generalization performance and recognition efficiency than HMM, and the training time of the SVM is also significantly shorter than HMM [49]. The length normalized GPLDA is initially used for face recognition [50] and then widely used in i-vector speaker verification [51]. Although the GPLDA classifier can achieve significant achievements in speech recognition [37], it cannot be applied well in the acoustic gesture recognition. The performance of GPLDA degrades at an increasing rates as speech duration decreases, when the acoustic signal utterance duration is 4 seconds, the equal error rate (EER) can soar to 17.38% [52], which will bring a relatively high equal error rate on gesture recognition system because the duration of input gesture is within 3 seconds or less. In contrast, SVM has fast training speed and high recognition accuracy, it does not rely on the length of the sound signal, which is suitable for implementation on mobile phones.

2) DEEP LEARNING

Deep learning methods convert acoustic signal to a series of spectrograms and utilize the latent and sophisticated image features to recognize hand gestures, for instance, deep neural network (DNN), deep recurrent neural network (DRNN). DNN increases the number of layers of neural network and hidden layers in order to improve the accuracy of recognition. DRNN combines deep neural network and recurrent neural network, it considers time sequence of different sequence lengths, so that it can extract the time information effectively and achieve more fine-grained identification. WordRecorder [53] refines original acoustic signals into normalized spectrograms and utilizes deep neural network models to recognize handwriting, it eventually achieves 81% accuracy rate on the smartwatch. Although the deep learning approaches based on neural network have been widely used in gesture recognition and handwriting classification, the real-time implementation on smartphone with high accuracy in noise and mismatched environment is not adequately addressed. The reason behind this phenomenon is that the neural network architectures are too large and deep in order to pursue high accuracy, this will sacrifice the real-time implementation of the system in practice [54]. Besides, the overfitting problem easily occurs in neural network model for the lack of massive training set [55]. Furthermore, if the ambient noise signal contaminates the original spectrogram, this recognition accuracy of deep learning will be greatly reduced. Some researchers explore applying the deep learning approaches on smartphone in real-time, they have to create complicated multi-threading technology [56]. Compared to the deep learning approaches, our system can run on smartphone efficiently in real-time with relatively low computation complexity. With only a small training set, the system can achieve a satisfactory accuracy in noise and mismatched environment.

VI. RELATED WORK

We classify existing work into (i) device-based gesture recognition and (ii) device-free gesture recognition. The device-based gesture recognition demands individuals to use tactile and haptic devices such as wireless sensors, cameras or accelerometers to support it, while device-free gesture recognition doesn't have the limits. The individuals can interact with smart devices freely without having to wear additional wearable devices.

A. DEVICE-BASED GESTURE RECOGNITION

Accelerometer sensor is often used for gesture sensing, for it can measure proper acceleration (''g-force'') from vibrations and the gravity. Accelerometer can sense the activities of human and capture the motion trajectory information precisely for recognizing gestures [57], [58]. Surface electromyography (SEMG) sensors provide another potential technology for gesture sensing. It has strong capability in capturing subtle movement such as wrist and finger movements, for example, with a wearable gesture sensing device (embedded with a three-axis accelerometer and four SEMG sensors) worn on the forearm, PG Scholar [59] is able to manipulate a mobile phone using 19 predefined gestures. AAMouse [60] uses the frequency shifts to estimate the velocity and track hand movement by smartphone, it calculates the distance in real time to locate the smartphone. Spartacus [61] leverages a novel acoustic technique based on the Doppler shift to estimate the device's moving direction and be interactive with devices through a pointing gesture. CAT [62] develops a distributed FMCW and combines it with the Doppler shift and IMU sensors over multiple time intervals to track and locate the moving object. RF-IDraw [63] can estimate the finger trajectory with high resolution and low ambiguity via 8 RFID antennas and the wearable RF tag.

However, all these systems demand the user to wear the interface devices, which is not convenient for VR/AR applications. Contact-based gesture recognition has many limits and is not adaptable to new users.

B. DEVICE-FREE GESTURE RECOGNITION

Except for the device-based gesture recognition schemes we discussed above, there are some device-free gesture recognition algorithms.

1) VISION-BASED

A great deal of the research into object tracking and gesture recognition is based on computer vision and vision processing technology. For example, Microsoft Kinect [1] uses depth sensor as well as camera to recognize a wide range of gestures, Wii [2] uses infrared cameras to track movement. LeapMotion [3] uses sophisticated vision technique similar to Kinect to track object.

However, these approaches are not suitable in smartphones for VR/AR applications since existing smartphones have limited computation power, energy, and sensing capability (i.e., no depth sensors). As a consequence, audio based interaction method is more attractive than vision to support smartphone-based VR by using the existing speaker and microphone on the phone, which will reduce energy

consumption significantly. As [64] reports, acoustic sensing only consumes 20% energy compared to vision-based object recognition.

2) RF-BASED

Recently, RF-based device-free object tracking algorithms in smart home and office environment have been widely investigated [65]–[70]. WiTrack [65] and WiTrack 2.0 [66] apply Frequency Modulated Continuous Wave (FMCW) to track a user's hand in high accuracy, WiDraw [69] estimates angle of arrival (AoA) using CSI and achieves a median tracking error of 5 cm using 25 WiFi access points (APs), mTrack [70] and Soli [68] use 60 GHz signals for gesture recognition.

Although these algorithms achieve considerably high performance, it is hardly to apply these algorithms in smartphones. For example, the FMCW signal used in WiTrack [65] and WiTrack 2.0 [66] needs customized hardware which can sweep the channel in nearly 2 GHz bandwidth. The large amount of AP used in WiDraw [69] limits its applicability. And WiDeo [67] uses WARP platform which is not readily available on the market. Moreover, 60GHz based gesture recognition schemes requires significant extra hardware for sending, receiving, and processing signals in real-time. Comparing with these RF-based applications, our system only requires the embedded microphone on the smartphone.

3) ACOUSTIC-BASED

Both LLAP [10] and FingerIO [12] track the finger movement using the reflected audio from a mobile phone. LLAP develops a phase based tracking while FingerIO uses OFDM symbol based movement detection. In addition, Strata *et al.* extract the path associated with the finger movement and track its phase change instead of using the mixed signals [13]. Although these schemes can track the finger movement in high accuracy, they cannot support gestures that require two or more fingers such as zoom in and out. On the other hand, WritingHacker [71] leverages the embedded microphone on the phone to snoop the victim's input gesture but it suffers from the poor recognition accuracy, and its performance is inevitably vulnerable to the ambient noise. UltraGesture [14] utilizes channel impulse response (CIR) based on ultrasonic finger motion perception to recognize human hand gesture, it identifies 12 hand postures with an average accuracy greater than 97%. WritePad [27] realizes consecutive number writing on the hand with a hybrid convolutional neural network model, the accuracy of number recognition is over 95%. Ipanel [24] is another system which utilizes passive acoustic sensing for gesture recognition and handwriting, it recognizes the finger movement and maintains 91.3% accuracy. iPand [55] adopts similar method and enables finger gesture input on the skin, it can classify 12 gestures with an overall accuracy of 83.8%.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new algorithm for gesture recognition using passive acoustic sensing and implemented it on COTS Android phones. We validated its effectiveness and robustness via comprehensive experiments. Comparing with our previous work "*SoundWrite*" [19], we apply a robustness signal segmentation scheme instead of detecting touch peak and end peak. Besides, we combined more advanced acoustic feature such as "MFCC", "CFCC", "STE" and "ZCR" to characterize the input acoustic signal. This fusion algorithm contributes to a sensible improvement (around 30%) accuracy. The results show that our algorithm can achieve 93.2% gesture recognition accuracy with 7 typical gestures, and it is robust in the presence of ambient noise and mismatched conditions.

We will incorporate more complicated gestures and improve the accuracy with deep learning algorithms such as CNN or DNN in the future. Besides, we will consider leveraging ultrasonic to detect the precise sliding position and direction for various gestures via multi-phone. Furthermore, we are to investigate how to decrease the memory consumption and make the system fast for real-time process to some extent.

REFERENCES

[1] *Microsoft X-box Kinect*. Accessed: 2020. [Online]. Available: http://xbox.com
[2] *Wii*. Accessed: 2020. [Online]. Available: https://www.nintendo.com/switch/
[3] *Leap Motion*. Accessed: 2020. [Online]. Available: https://www.ultraleap.com/
[4] *Google Project Soli*. Accessed: 2020. [Online]. Available: https://atap.google.com/soli
[5] C. Harrison and S. E. Hudson, "Scratch input: Creating large, inexpensive, unpowered and mobile finger input surfaces," in *Proc. ACM Symp. User Interface Softw. Technol.*, 2008, pp. 205–208.
[6] B. Kellogg, V. Talla, and S. Gollakota, "Bringing gesture recognition to all devices," in *Proc. Usenix Conf. Netw. Syst. Design Implement.*, 2014, pp. 303–316.
[7] L. Chan, R. H. Liang, M. C. Tsai, K. Y. Cheng, C. H. Su, M. Y. Chen, W. H. Cheng, and B. Y. Chen, "Fingerpad: Private and subtle interaction using fingertips," in *Proc. ACM User Interface Softw. Technol. Symp.*, 2013, pp. 255–260.
[8] J. Gummeson, B. Priyantha, and J. Liu, "An energy harvesting wearable ring platform for gestureinput on surfaces," in *Proc. 12th Annu. Int. Conf. Mobile Syst., Appl., Services*, 2014, pp. 162–175.
[9] K. Y. Chen, S. N. Patel, and S. Keller, "Finexus: Tracking precise motions of multiple fingertips using magnetic sensing," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2016, pp. 1504–1514.
[10] W. Wang, A. X. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," in *Proc. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 82–94.
[11] S. Gupta, D. Morris, S. Patel, and D. Tan, "SoundWave: Using the Doppler effect to sense gestures," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2016, pp. 1911–1914.
[12] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "Fingerio: Using active sonar for fine-grained finger tracking," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2016, pp. 1515–1525.
[13] S. Yun, Y.-C. Chen, H. Zheng, L. Qiu, and W. Mao, "Strata: Fine-grained acoustic-based device-free tracking," in *Proc. 15th Annu. Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2017, pp. 15–28.
[14] K. Ling, H. Dai, Y. Liu, and A. X. Liu, "UltraGesture: Fine-grained gesture sensing and recognition," in *Proc. 15th Annu. IEEE Int. Conf. Sens., Commun., Netw. (SECON)*, Jun. 2018, pp. 1–9.
[15] K. Sun, T. Zhao, W. Wang, and L. Xie, "Vskin: Sensing touch gestures on surfaces of mobile devices using acoustic signals," in *Proc. 24th Annu. Int. Conf. Mobile Comput. Netw.*, 2018, pp. 591–605.
[16] Y. Zou, Q. Yang, Y. Han, D. Wang, J. Cao, and K. Wu, "AcouDigits: Enabling users to input digits in the air," in *Proc. IEEE PerCom*, Mar. 2019, pp. 313–321.

[17] J. Wang, K. Zhao, X. Zhang, and C. Peng, "Ubiquitous keyboard for small mobile devices: Harnessing multipath fading for fine-grained keystroke localization," in *Proc. Int. Conf. Mobile Syst., Appl., Services*, 2014, pp. 14–27.

[18] T. Zhu, Q. Ma, S. Zhang, and Y. Liu, "Context-free attacks using keyboard acoustic emanations," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2014, pp. 453–464.

[19] M. Zhang, P. Yang, C. Tian, L. Shi, S. Tang, and F. Xiao, "SoundWrite: Text input on surfaces through mobile acoustic sensing," in *Proc. Int. Workshop Exper. Design Implement. Smart Objects*, 2015, pp. 13–17.

[20] H. Yin, A. Zhou, L. Liu, N. Wang, and H. Ma, "Ubiquitous writer: Robust text input for small mobile devices via acoustic sensing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5285–5296, Jun. 2019.

[21] M. Zhang, L. Ping, P. Yang, X. Jie, and T. Chang, "Poster: Sonicnect: Accurate hands-free gesture input system with smart acoustic sensing," in *Proc. 14th Annu. Int. Conf.*, 2016, p. 91.

[22] Q. Li and Y. Huang, "An auditory-based feature extraction algorithm for robust speaker identification under mismatched conditions," *IEEE Trans. Audio, Speech, Language Process.*, vol. 19, no. 6, pp. 1791–1801, Aug. 2011.

[23] *SVM*. Accessed: 2020. [Online]. Available: https://en.wikipedia.org/wiki/Support_vector_machine

[24] M. Chen, P. Yang, J. Xiong, M. Zhang, Y. Lee, C. Xiang, and C. Tian, "Your table can be an input panel: Acoustic-based device-free interaction recognition," in *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, 2019, vol. 3, no. 1, p. 3.

[25] J.-C. Junqua, B. Reaves, and B. Mak, "A study of endpoint detection algorithms in adverse conditions: Incidence on a DTW and HMM recognizer," in *Proc. 2nd Eur. Conf. Speech Commun. Technol.*, 1991, pp. 1371–1374.

[26] R. Narayanam, "Voiced and unvoiced separation in speech auditory brainstem responses of human subjects using zero crossing rate (ZCR) and energy of the speech signal," *Int. J. Eng. Sci. Res. Technol.*, pp. 370–380, 2017.

[27] M. Chen, P. Yang, S. Cao, M. Zhang, and P. Li, "WritePad: Consecutive number writing on your hand with smart acoustic sensing," *IEEE Access*, vol. 6, pp. 77240–77249, 2018.

[28] *MFCC*. Accessed: 2019. [Online]. Available: https://en.wikipedia.org/wiki/Mel-frequency_cepstrum

[29] S. Nakagawa, L. Wang, and S. Ohtsuka, "Speaker Identification and Verification by Combining MFCC and Phase Information," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 4, pp. 1085–1095, May 2012.

[30] A. Shafik, S. M. Elhalafawy, S. Diab, B. M. Sallam, and F. A. El-Samie, "A wavelet based approach for speaker identification from degraded speech," *Int. J. Commun. Netw. Inf. Secur.*, vol. 1, no. 3, p. 52, 2009.

[31] S. Ganapathy, J. Pelecanos, and M. K. Omar, "Feature normalization for speaker verification in room reverberation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 4836–4839.

[32] Q. Li, "An auditory-based transfrom for audio signal processing," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, Oct. 2009, pp. 181–184.

[33] *MFCC Java*. Accessed: 2012. [Online]. Available: http://download.csdn.net/download/liuyuew/4318634

[34] *LIBSVM*. Accessed: 2020. [Online]. Available: https://en.wikipedia.org/wiki/LIBSVM

[35] R. J. Vogt, "Automatic speaker recognition under adverse conditions," Ph.D. dissertation, Queensland Univ. Technol., Brisbane, QLD, Australia, 2006.

[36] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," *Int. Speech Commun. Assoc.*, pp. 213–218, 2001.

[37] A. K. H. Al-Ali, D. Dean, B. Senadji, V. Chandran, and G. R. Naik, "Enhanced forensic speaker verification using a combination of DWT and MFCC feature warping in the presence of noise and reverberation conditions," *IEEE Access*, vol. 5, pp. 15400–15413, 2017.

[38] A. K. H. Al-Ali, B. Senadji, and V. Chandran, "Hybrid DWT and MFCC feature warping for noisy forensic speaker verification in room reverberation," in *Proc. IEEE Int. Conf. Signal Image Process. Appl. (ICSIPA)*, Sep. 2017, pp. 434–439.

[39] J. Mcauley, J. Ming, D. Stewart, and P. Hanna, "Subband correlation and robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 956–964, Sep. 2005.

[40] X. Jing, B. Xiang, H. Yang, and P. Zhou, "Robust speaker verification using improved PNCC based on GMM-UBM," *Int. J. Autom. Power Eng*, vol. 4, pp. 14–19, 2015.

[41] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 7, pp. 1315–1329, Jul. 2016.

[42] M. T. Al-Kaltakchi, W. L. Woo, S. S. Dlay, and J. A. Chambers, "Study of fusion strategies and exploiting the combination of MFCC and PNCC features for robust biometric speaker identification," in *Proc. 4th Int. Conf. Biometrics Forensics (IWBF)*, Mar. 2016, pp. 1–6.

[43] R. Yao, Z. Zeng, and P. Zhu, "A priori SNR estimation and noise estimation for speech enhancement," *EURASIP J. Adv. Signal Process.*, vol. 2016, no. 1, p. 101, 2016.

[44] J. Li and D. You, "Enhanced speech based jointly statistical probability distribution function for voice activity detection," *Chin. J. Electron.*, vol. 26, no. 2, pp. 325–330, Mar. 2017.

[45] G.-H. Ding, X. Wang, Y. Cao, F. Ding, and Y. Tang, "Speech enhancement based on speech spectral complex Gaussian mixture model," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., (ICASSP)*, vol. 1, Oct. 2005, p. I-165.

[46] H. Ai, K. Tang, L. Han, Y. Wang, and S. Zhang, "DuG: Dual speaker-based acoustic gesture recognition for humanoid robot control," *Inf. Sci.*, vol. 504, pp. 84–94, Dec. 2019.

[47] Z. Wang, Y. Hou, K. Jiang, W. Dou, C. Zhang, Z. Huang, and Y. Guo, "Hand gesture recognition based on active ultrasonic sensing of smartphone: A survey," *IEEE Access*, vol. 7, pp. 111897–111922, 2019.

[48] B. Bansal, "Gesture recognition: A survey," *Int. J. Comput. Appl.*, vol. 139, no. 2, pp. 8–10, Apr. 2016.

[49] W. Cao, "Application of support vector machine based speech recognition technology in human-computer interaction technology," *Int. J.*, no. 5, p. 70, 2019.

[50] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Proc. IEEE 11th Int. Conf. Comput. Vis. (ICCV)*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.

[51] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Proc. INTERSPEECH 12th Annu. Conf. Int. Speech Commun. Assoc.*, Florence, Italy, Aug. 2011, pp. 249–252.

[52] A. Kanagasundaram, R. Vogt, D. Dean, S. Sridharan, and M. Mason, "i-vector based speaker recognition on short utterances," in *Proc. INTERSPEECH*, 2011, pp. 2341–2344.

[53] H. Du, P. Li, H. Zhou, W. Gong, G. Luo, and P. Yang, "Wordrecorder: Accurate acoustic-based handwriting recognition using deep learning," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2018, pp. 1448–1456.

[54] A. Sehgal, F. Saki, and N. Kehtarnavaz, "Real-time implementation of voice activity detector on arm embedded processor of smartphones," in *Proc. IEEE 26th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2017, pp. 1285–1290.

[55] S. Cao, X. He, P. Zhu, M. Chen, X. Li, and P. Yang, "IPand: Accurate gesture input with ambient acoustic sensing on hand," in *Proc. IEEE 37th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Nov. 2018, pp. 1–8.

[56] A. Sehgal and N. Kehtarnavaz, "A convolutional neural network smartphone app for real-time voice activity detection," *IEEE Access*, vol. 6, pp. 9017–9026, 2018.

[57] J.-S. Wang and F.-C. Chuang, "An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition," *IEEE Trans. Ind. Electron.*, vol. 59, no. 7, pp. 2998–3007, Jul. 2012.

[58] C. Zhu and W. Sheng, "Wearable sensor-based hand gesture and daily activity recognition for robot-assisted living," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 3, pp. 569–573, May 2011.

[59] 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.

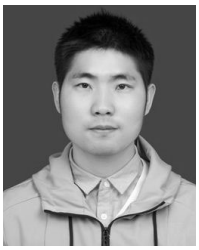
[60] S. Yun, Y. C. Chen, and L. Qiu, "Turning a mobile device into a mouse in the air," in *Proc. Int. Conf. Mobile Syst., Appl., Services*, 2015, pp. 15–29.

[61] Z. Sun, R. Bose, and P. Zhang, "Spartacus: Spatially-aware interaction for mobile devices through energy-efficient audio sensing," *GetMobile, Mobile Comput. Commun.*, vol. 18, no. 4, pp. 11–14, 2015.

[62] W. Mao, J. He, and L. Qiu, "Cat: high-precision acoustic motion tracking," in *Proc. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 69–81.

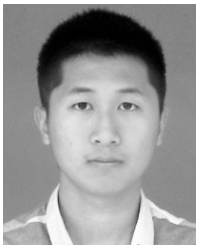
[63] J. Wang, D. Vasisht, and D. Katabi, "RF-IDraw: Virtual touch screen in the air using RF signals," in *Proc. ACM Conf. SIGCOMM*, 2015, pp. 235–246.

[64] K. G. Shin and Y. C. Tung, "Real-time warning for distracted pedestrians with smartphones," U.S. Patent 10 036 809, Jul. 31, 2018.
[65] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, "3D tracking via body radio reflections," in *Proc. 11th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2013, pp. 317–329.
[66] F. Adib, Z. Kabelac, and D. Katabi, "Multi-person localization via RF body reflections," in *Proc. USENIX Conf. Netw. Syst. Design Implement.*, 2015, pp. 279–292.
[67] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti, "WiDeo: Fine-grained Device-free Motion Tracing using RF backscatter," in *Proc. USENIX Conf. Netw. Syst. Design Implement.*, 2015, pp. 189–204.
[68] J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev, "Soli: Ubiquitous gesture sensing with millimeter wave radar," *ACM Trans. Graph.*, vol. 35, no. 4, p. 142, 2016.
[69] L. Sun, S. Sen, D. Koutsonikolas, and K. H. Kim, "Widraw: Enabling hands-free drawing in the air on commodity WiFi devices," in *Proc. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 77–89.
[70] T. Wei and X. Zhang, "mTrack: High-precision passive tracking using millimeter wave radios," in *Proc. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 117–129.
[71] T. Yu, H. Jin, and K. Nahrstedt, "Writinghacker: Audio based eavesdropping of handwriting via mobile devices," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2016, pp. 463–473.



GAN LUO received the B.S. degree from the College of Computer Science and Electronics, Hunan University, China, in 2011, and the M.S. degree in computer science and technology from the Army Engineering University of PLA, China, in 2018. His current research interests include acoustic sensing, UAV, as well as object detection and tracking.

PANLONG YANG received the B.S., M.S., and Ph.D. degrees in communication and information system from the Nanjing Institute of Communication Engineering, China, in 1999, 2002, and 2005, respectively. He is currently a Professor with the College of Computer Science and Technology, University of Science and Technology of China. His research interests include wireless mesh networks, wireless sensor networks, and cognitive radio networks. He is a member of the IEEE Computer Society and the ACM SIGMOBILE Society.



MINGSHI CHEN received the B.S. degree from Zhejiang University, China, in 2016, and the M.S. degree in computer science and technology from the Army Engineering University of PLA, China, in 2018. His current research interests include acoustic sensing and artificial intelligence.

PING LI received the B.S. degree in applied chemistry from the College of Chemistry and Chemical Engineering, Anhui University, China, in 2007, and the M.S. degree in electronic and communication engineering from the College of Communications Engineering, PLA University of Science and Technology, China, in 2016. He is currently pursuing the Ph.D. degree with the Army Engineering University of PLA. His current research interests include indoor localization, software radio systems, and interference management.