AcouDigits: Enabling Users to Input Digits in the Air

Yongpan Zou*, Qiang Yang*, Yetong Han*, Dan Wang*, Jiannong Cao‡, Kaishun Wu*†
*College of Computer Science and Software engineering, Shenzhen University
†PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen, China
‡Department of Computing, Hong Kong Polytechnic University
{yangqiang2016,hanyetong2017,wangdan9}@email.szu.edu.cn
{yongpan,wu}@szu.edu.cn, csjcao@comp.polyu.edu.hk

***Abstract*—Recently, wearable devices have become increasingly popular in our lives because of their neat features and stylish appearance. However, due to the tiny size, it is inconvenient for users to interact with a device using conventional methods, especially for text entry. Although some methods have been proposed to handle this problem, they have different limitations and are not applicable to many existing mobile devices. As a result, we take the first step to propose a digits-entry system, *i.e.*, AcouDigits, in which digits can be entered in the air using a finger without taking help from any additional hardware. We implement AcouDigits on two commercial devices and conduct experiments to evaluate its performance in recognizing ten basic digits. Experimental results show that AcouDigits can achieve average accuracies of 91.7% and 87.4% in recognizing basic digits and 26 English alphabets, respectively.**

***Index Terms*—Text entry, Acoustic signals, Wearable devices**

I. Introduction

In the most recent years, wearable devices have burst into enormous prosperity owing to the technological progress in integrated circuit (IC), high-performance chip, energy-intensive battery, *etc.*. According to a forecast, an explosively growing number of people tend to equip themselves with at least one of the wearable devices (*e.g.*, smartwatch, smartband and smartglass) [1]. Compared with other smart devices, wearables are mostly with tiny sizes to ensure comfortable user experience and enhance aesthetic appearance. However, it incurs much inconvenience for users to interact with devices, especially in the case of text entry. Because of the tiny screens, interactive methods, such as soft keyboards cannot work efficiently in present wearable devices although these are widely utilized in conventional mobile devices. In such a situation, even dialing telephone numbers is a labor-intensive task. Another shortcoming of soft keyboard is that it requires a user to touch the screen with fingers. When the fingers of a user are wet or dirty, this method no longer works.

Researchers have explored various methods for text entry on tiny devices. Among them, speech recognition is a promising choice that enables users to convey commands and sentences by speaking. Nevertheless, this method possesses several intrinsic shortcomings, such as privacy leakage in public places, performance degradation in noisy environments, and inconvenience in certain occasions (*e.g.*, in a conference or in a library). Among other existing text-based HCI techniques, some are based on radio-frequency (RF) signals received from RFID or Wi-Fi networks [2]–[5], and some are based on inertial sensors [6]–[10]. However, RF-based technique needs specialized equipment (*e.g.*, RFID readers and tags, multi-antenna Wi-Fi transceiver, *etc.*.) and/or requires users to attach additional hardware (tags) on them, which make this method inapplicable to mobile devices. Besides, the inertial sensor-based scheme requires a user to hold the device with hand(s) or wear an additional device while writing text in the air. Consequently, these equipment-centric schemes appeared not to be appealing to users. This observation motivates us to pursue in this direction of research. Now, in this context, one may ask a crucial question, *i.e.*, *can we input digits in a device-free manner without any additional hardware?*

As a preliminary attempt in responding to this question, in this paper, we propose a system named AcouDigits, which enables a user to enter digits using his/her finger without touching the screen and wearing any additional devices. In our system, acoustic sensors (*e.g.*, microphone and speaker), pervasively embedded in smart devices, are fully utilized to emit and receive high-frequency signals. To input a digit, a user just needs to write it in the air near a device as if a virtual keyboard was placed there. Actually, acoustic signals have already been utilized to build human-device interaction systems in a device-free manner. For example, AirLink [11], Dolphin [12], Soundwave [13] and FingerIO [14] are some closely related works in this context. However, in AirLink, Dolphin and Soundwave, a user can interact with the device using coarse hand gestures that utilize the Doppler effect of acoustic signals. Nevertheless, the aim of recognizing digits in the air written by only a finger exposes several challenges. With high precision, FingerIO solves the fine-grained finger motion tracking problem. However, in order to recognize digits, FingerIO requires multiple microphones to track finer motions in 2D or 3D space, which are not available in most existing commercial devices, especially wearable devices. To the best of our knowledge, AcouDigits is the first work that makes use of acoustic signals to enable text entry, and is suitable to be deployed on the most existing commercial devices. We claim that AcouDigits outperforms the previous works in terms of two main aspects, which are as follows. First, previous works mainly focused on recognizing several

978-1-5386-9148-9/19/$31.00 ©2019 IEEE

predefined coarse gestures, while AcouDigits is able to recognize a larger set of objects with finer-grained granularity. Second, some previous works require a user to carry a device using his/her hand(s), while AcouDigits works in a device-free manner.

The underlying principle of AcouDigits is straightforward, which is briefly described as follows. When a user writes a digit in the air, the received echo wave is altered due to the multi-path effect. Owing to the unique writing path of a specific character or number, the multi-path pattern reflected in received signals provides the potential intuition of text recognition. However, the implementation of this system exposes a key challenge, which is how to extract, and then differentiate the minute signal patterns of each written digit is a question. To overcome these challenges, we carefully design the entire data processing module and adopt a two-layer feature engineering scheme to extract patterns hidden in signals. To examine the effectiveness of AcouDigits, we implement the system on Android platform (Samsung Galaxy Note 5), and conduct comprehensive experiments in different settings. The experimental results show that AcouDigits can recognize basic digits and English alphabets with average accuracies of 91.7% and 87.4%, respectively. The main contributions of our work can be summarized as follows.

- We make use of embedded sensors to design a text-entry scheme for tiny devices. By carefully designing signal processing methods, we overcome the technical challenges of removing noise interference and detecting writing events. By combing the physical domain knowledge with machine learning, we achieve favorable digits recognition performance.
- We implement such an scheme on mobile devices and conduct experiments to evaluate its performance under different settings. Experimental results show that our system can recognize basic digits and alphabets with high accuracy.

The remainder of this paper is organized as follows. Section II introduces the related works. In Section III, we present the specific technical design of AcouDigits. Section IV introduces the experimental design and performance evaluation. Last but not the least, we conduct discussion in Section V and conclude the paper in Section VI.

II. Related Work

AcouDigits is closely related previous works in the following aspects.

A. Commercial text-entry approaches

As mobile devices are increasingly popular in our daily lives, it attracts much attention to improving the interaction experience and convenience of entering texts to these devices. Prior to our work, various text-entry systems have been proposed towards this goal. Soft keyboard is almost the most common interface for text entry in present smartphones, tablets and other kinds of mobile devices. This method possesses advantages of low cost, efficiency and convenience. However,

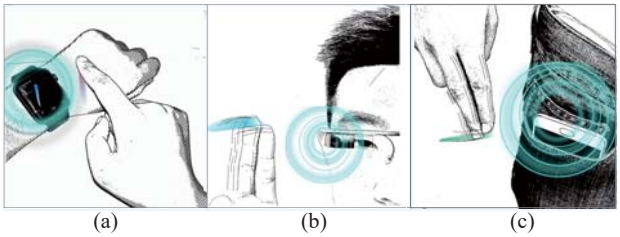


Fig. 1. Possible scenarios where AcouDigits can be deployed. AcouDigits cannot only be deployed on wearable devices, such as (a) smartwatches and (b) smartglasses to deal with the inconvenience caused by small screens; but also on (c) mobile devices, such as smartphones and tablets to handle the cases where hands are wet or oil-scalded.

for devices with tiny screen sizes such as smartwatches and smartglasses, it is usually difficult for users to enter texts efficiently and conveniently. Some improvements such as [15], [16] have been made to raise the inputting efficiency on smartwatches and smartphones. Another promising and widely applied technique for text entry is speech recognition, owing to its high accuracy and favorable experience. Within this scope, Siri [17] and Cortana [18] are the outstanding speech assistants. However, it is to be admitted that speech recognition is not perfect in all cases. For example, the recognition performance will be affected by the surroundings and degrade in a noisy environment. Moreover, using speech assistant in public occasions has the risk of privacy leakage and brings about awkward feelings to some users.

B. Sensor and RF-based text-entry systems

Besides the above commercial techniques, researchers have also proposed some other schemes to deal with the problems, namely sensor-based scheme [6], [7], [10] and radio frequency (RF) based scheme [2]–[5]. In [6], the accelerometer embedded in a smartphone is utilized to recognize the characters when a user writes texts in the air with a smartphone. In [7], [10], various sensors such as accelerometer, proximity sensor and distance sensors have been fused to design a middleware to enter texts to other devices. However, compared with AcouDigits, these existing works have either shortcomings as follows: 1) they need additional devices except the device to interact with; 2) they require the user to carry the device in his/her hand when entering texts. In the RF-based scheme, Wi-Fi, RFID and 60 GHz transceivers have been adopted to design high-precision motion tracking systems [3], [4] or text-input system [2], [19]. Nevertheless, RF-based systems require deploying sophisticated RF equipment and are not applicable for present mobile devices. In contrast, with all the aforementioned approaches, AcouDigits provides a method for text input without any additional device and does not require the user to carry or wear any equipment in hand or on body.

C. Acoustic signal-based HCI

Acoustic sensors, namely, microphone and speaker, have been widely used to design interactive systems in mobile devices. Some researchers make use of the Doppler effect of

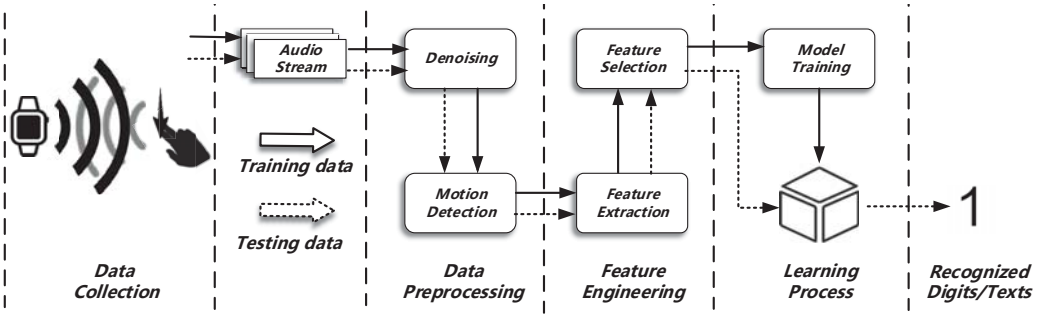


Fig. 2. The functional flow diagram of AcouDigits.

acoustic signals to sense gestures [11], [13], [20] and track motions [21]–[24]. However, all these systems except [11], [13] need the user to carry a smartphone in hand and perform some gestures, which results in great inconvenience to a user. Moreover, the techniques utilized in these works can not be applied to recognize texts as demonstrated in our application scenarios. Although [11] and [13] can sense in-air gestures with acoustic signals, they both only utilize Doppler shift caused by hand gestures as the feature to distinguish a limited number of coarse gestures such as 'PULL' and 'FLICK'. In order to recognize texts written with a finger, a finer recognition granularity is required. The most closely related work to AcouDigits are [14], [25], [26], in which device-free 2D finger motion tracking systems are designed with mm-level precision. However, there are dominant differences between our work and them. Fist, those works focus on motion tracking instead of text entry. To recognize texts, their systems need additional extensions. More importantly, in order to track 2D motion of a finger, those works require multiple speaker-microphone pairs to be equipped in a mobile device. Nevertheless, limited by the cost and tiny size, an overwhelming majority of present smartphones and smartwatches can not fulfill this requirement. Compared with them, AcouDigits only relies on one pair of speaker and microphone, which makes it be easily implemented on almost all commercial smart devices.

III. System Description

Fig. 2 provides an overview of AcouDigits, in which we describe the system at the detailed functional level. To be specific, the speaker embedded in a smart device emits sin wave modulated acoustic signals which are bounced off by a finger that is writing digits near that device. The microphone(s) of the same device receives the bounced acoustic signals. The received signals are first denoised to filter out interfering noise and magnify the signal components of interest. Then, writing activities are detected and the corresponding signal segments are extracted. On the extracted segments, feature engineering techniques, *i.e.*, feature extraction and feature selection are applied in a consecutive manner. In the last stage, learning models, namely K-nearest neighbor (KNN), support vector machine (SVM) and artificial neural network (ANN), are trained with the processed data. We tune the parameters of the learning models in such a way that the recognition performance is enhanced. From the high level, AcouDigits is composed of three main modules, which are data preprocessor, feature engineering scheme and learning process. The detailed functional description of each module is provided in the following.

A. Data Preprocessing

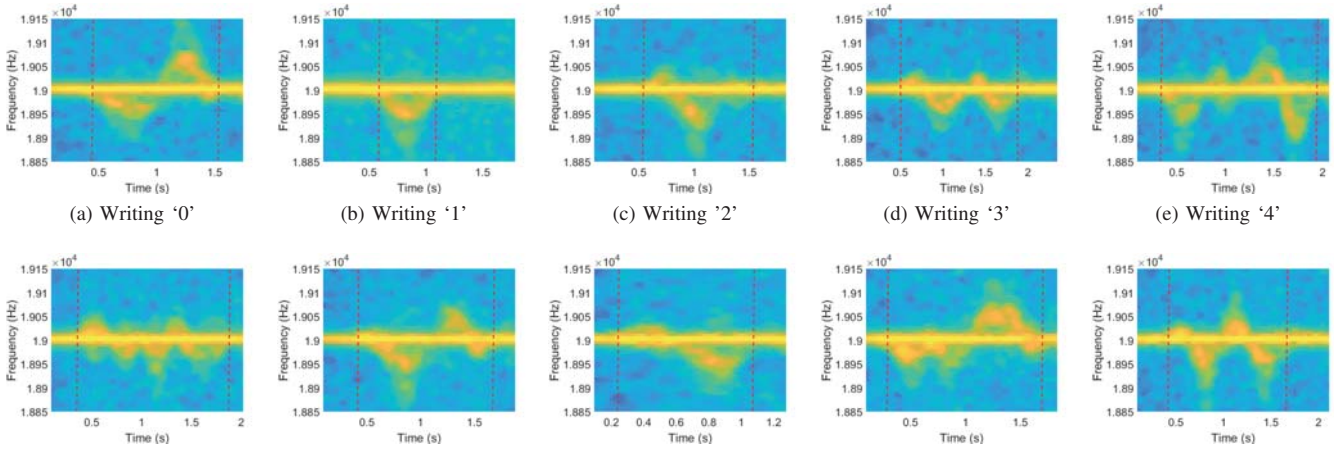
After sampling raw audio signals, the received data is fed into a preprocessing module which removes noise and enhances SNR. To be precise, in this stage, we first denoise the raw audio signals, and then detect the finger motion activities from the denoised data. We utilize high-frequency acoustic signals (19 KHz), since it is inaudible to most human beings. For this reason, we only focus on the received signals of a certain band, which is centered around 19 KHz along with frequency shifts caused by finger motions. Signal components beyond this band are regarded as noise, and filtered out by a band-pass filter. To determine the parameters of the filter, it is required to determine the amount of frequency shifts when a finger starts writing. Considering a case where the finger moves at 1 m/s velocity, the resultant maximal frequency shift is about 112 Hz which is determined by:

$$\Delta f = f_0 \cdot |1 - \frac{v_s \pm v_f}{v_s \mp v_f}|$$

where f_0, v_s and v_f represent the frequency of emitted signals, the speed of sound and the velocity of finger motion, respectively. Based on this, we employ a 6-order Butterworth bandpass filter, where the passband is set to be $[18850, 19150]$ Hz, in order to remove irrelevant interference such as background noises and human voices to a great extent. Butterworth filter is designed to have maximum flat frequency response in the pass band and roll off towards zero in the stop band, which ensures the fidelity of signals in target frequency range while removing out-band noises greatly [27].

B. Event Detection

Followed by the denoising process, it is required to detect the writing activity, and extract the corresponding signal segments. Different from most of the previous works that analyze the original data, we decide to detect the writing activities from the spectrogram of original data. The key reason of such



(a) Writing '0' (b) Writing '1' (c) Writing '2' (d) Writing '3' (e) Writing '4'

(f) Writing '5' (g) Writing '6' (h) Writing '7' (i) Writing '8' (j) Writing '9'

Fig. 3. Illustrations of spectrograms and writing activity detection.

decision is that the finger motion causes frequency shift in the frequency domain. After transforming each frame of signal into the frequency domain, it is straightforward to detect the start and end points of each writing activity. Technically, we apply a sliding Hanning window on the signal sequence and obtain a series of audio frames. The reason why a Hanning window is applied is to reduce spectrum leakage in the following Fourier transform [28]. After that, we apply Fast Fourier Transform (FFT) on each frame to obtain the spectrum [27], and then check whether there exist frequency shifts around 19 KHz band. In detail, we set the frame length and nFFT (the length of FFT) to be 8192 sample points (about 0.186 s) and the overlapping length between consecutive frames to half of the frame length, by considering the trade-off between time resolution and frequency resolution. After that, we can obtain the spectrogram of signals in each frame. To decide whether the frame is active or not, we check the frequency shift of the obtained spectrogram at each frequency. When there exists a threshold number (we set this threshold to 4) of continuous frequency shifts, the frame is asserted to be active. However, the frequency shifts can be caused by some random movements instead of writing activities. Hence, after collecting a series of active frames, we apply filtering on each frame due to the fact that the duration of a writing activity exceeds a certain threshold. The results of detecting writing different digits based on spectrograms are given in Fig. 3. As we can see, when a user writes '0', his/her finger sweeps through the microphone and back, which makes the resultant spectrum emerge like a sine wave. As for digit '1', a user writes it with finger going away from microphone directly. As a result, a 'valley' appears in the spectrogram. However, for digits '2' and '7', '3' and '9', they spectrograms exhibit similar patterns, which makes it impossible to differentiate them only with frequency-domain features. After detecting writing events, we further apply a notch filter on each obtained signal sequence in order to remove 19 KHz frequency components.

TABLE I
EXTRACTED ACOUSTIC FEATURES IN ACOUDIGITS

Feature domain	Feature	Description
Time domain	Root mean square (RMS)	The energy in an acoustic frame
	Zero crossing rate (ZCR)	The point where acoustic samples change signs
	ATR	The average value of top k RMSs
	Above α-mean ratio (AMR)	The ratio of high-energy frames in a window
	AC	Auto-correlation coefficients
Frequency domain	Spectral entropy (SE)	The flatness indicator of acoustic spectrum shape
	Spectral flux (SF)	The stability reflector of acoustic events
	Spectral rolloff (SR)	Indicator of a frame's spectral energy distribution
	Spectral centroid (SC)	The balance point of the spectral energy distribution

C. Feature Engineering

Since the quality of a feature set has a great influence on the performance of machine learning models, feature engineering is required for the learning models of AcouDigits. In the following, we provide detailed description of this technique, which includes both feature extraction and feature selection mechanisms.

1) Feature Extraction: At this stage, we need to determine a set of features to build effective learning models for AcouDigits. However, to extract effective features, it usually requires profound domain knowledge and deep insight to the solution structure of the problem. For this, we select some widely-used acoustic features in both time and frequency domains, which are shown in Table I. Let $S(i), i = 1, 2, .., n$ represents

a frame of acoustic signal, where n is the number of data points. To obtain better knowledge of these features, we give their definitions with brief introductions as follows.

- **Root mean square (RMS)** This feature represents the energy in an acoustic frame, which can be calculated as:

$$RMS = \sqrt{\frac{\sum_{i=1}^{n} S^2(i)}{n}} \tag{1}$$

- **Zero crossing rate (ZCR)** ZCR describes the rate of sign-changes along a signal. This feature can be defined as:

$$ZCR = \frac{1}{2(n-1)} \sum_{i=1}^{n-1} |sgn(S(i)) - sgn(S(i+1))| \tag{2}$$

where sgn is the sign function.

- **Average value of top k RMSs (ATR)** This feature denotes the average RMS value of the first k frames with the most energy in a window w. ATR is calculated as:

$$ATR(k, w) = \frac{\sum_{i=1}^{k} RMS(f_i)}{k} \tag{3}$$

- **Above α-mean ratio (AMR)** AMR is a feature that can describe the ratio of the high-energy frames in a window w, which is defined as:

$$AMR(\alpha, w) = \frac{1}{m} \mathfrak{B}[RMS(f_i) > \alpha \cdot \overline{RMS}(w)] \tag{4}$$

where $\mathfrak{B}$ is the Boolean function.

- **Auto-correlation coefficients (AC)** This feature calculates the randomness (or periodicity) of a signal. Let $y(n)$ denotes a frame S, $y(n-l)$ represents the signal that S delayed l samples. AC is defined by:

$$AC = \frac{\frac{1}{n} \sum_{l=1}^{n-l} (y(n) - \bar{y})(y(n-l) - \bar{y})}{\sigma_y^2} \tag{5}$$

where $\bar{y}$ is the mean value of $y(n)$ (*i.e.*, frame S), and σ_y^2 refer to the sample variance of $y(n)$. It should be noted that AC is a sequence as same length as frame S.

In the following, we introduce features in frequency domain as shown in Table I. $P_i, i = 1, 2, ...n$ denotes the normalized magnitude of the i-th frequency bin obtained by performing FFT on a frame.

- **Spectral entropy (SE)** SE reflects the complexity of a system, which can be obtained as:

$$SE = -\sum_{i=1}^{n} p_i log_2 p_i \tag{6}$$

- **Spectral flux (SF)** This feature describes how drastically the acoustic signal is changing between frames, which is defined as the L2-norm of the spectral amplitude difference of two adjacent frames.

$$SF = \sum_{i=1}^{n} (p_t(i) - p_{t-1}(i))^2 \tag{7}$$

where $p_t(i)$ and $p_{t-1}(i)$ refer to spectral magnitude at ith frequency bin of current and previous frame, respectively.

- **Spectral rolloff (SR)** SR indicates the skewness of the spectral amplitude distribution, which can be calculated by the frequency bin below that it contains percent of the total spectral magnitude.

$$SR = max(h | \sum_{i=1}^{h} p(i) < \lambda \sum_{i=1}^{n} p(i)) \tag{8}$$

- **Spectral centroid (SC)** This feature reveals the balance point of the spectral energy distribution, it is the weighted mean of the frequencies and weighted by magnitudes of frequency bins.

$$SC = \frac{\sum_{i=1}^{n} i \times p(i)^2}{\sum_{i=1}^{n} p(i)^2} \tag{9}$$

In our specific design, we divide the entire acoustic sequence of each writing activity into frames, each of which contains 8192 samples (*i.e.*, 0.186 seconds), and extract the aforementioned features on each frame. Consequently, we obtain a set of discrete series corresponding to each writing activity. As we employ KNN, SVM and ANN as our learning models, we further conduct feature extraction in the second tier. To be specific, as for KNN model, we directly feed the extracted discrete series into the learning algorithm. In contrast, as for SVM and ANN models, we further extract statistical features including *mean value*, *variance*, *range*, *kurtosis* and *skewness* from the aforementioned discrete series.

2) Feature Selection: Feature selection is an important technique to filter out noisy and redundant features to improve learning performance and reduce computational cost. For our system, while building the learning models, we select features heuristically using 10-fold cross validation technique. The primary criterion of evaluating a feature set is the average accuracy rate (AAR) of recognizing different digits. When the AAR of different feature sets are almost similar, the next factor, *i.e.*, training overhead is taken into account. We randomly select a subset of features and evaluate its performance in terms of AAR and training overhead. We also observe that the feature set {AC, SC, SF} outperforms all the other feature sets, as shown in Fig 4, and is selected to train all models. Theoretically, AC is able to differentiate the signal reflected by finger writing with periodicity, like {3, 8}. SF describes the changeability of signals, which is an indicator of discriminating digits {1, 2, 3, 7} and {4, 5, 6, 8, 9}, since the latter have more complex writing patterns. SC describes the balanced point of spectral energy distribution and can be used to distinguish digits whose signal energy concentrates on different frequencies. In the second tier, we select {sum (Sum), variance (Var), mean (Mean)} as the feature set, to be utilized in SVM and ANN learning models, which can further demonstrate the variety of features over time.

D. Model Training

As mentioned above, the learning models that we have chosen for designing our system are KNN, SVM and ANN.

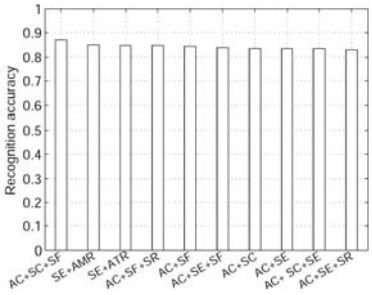


Fig. 4. AAR of different feature sets.

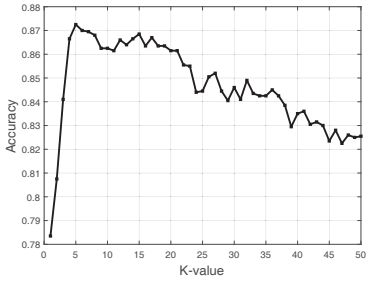


Fig. 5. Performance of different K-values.

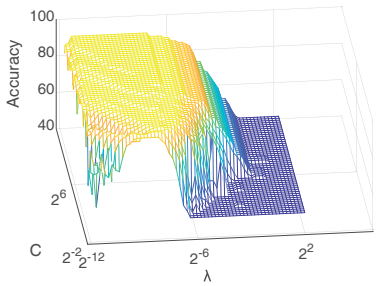


Fig. 6. Performance of different parameters of SVM.

TABLE II
PERFORMANCE OF DIFFERENT TRAINING FUNCTIONS

Training functions	trainlm	trainbr	trainbfg	trainrp	trainscg	traincgb	traincgf	traincgp	trainoss	traingdx	traingdm	traingd
Training accuracy	94.80%	98.60%	77.10%	86.90%	83.60%	81.50%	82.20%	84.00%	81.40%	78.60%	16.90%	6.90%
Testing accuracy	92.80%	90.00%	77.00%	87.70%	81.00%	80.70%	82.30%	83.30%	81.00%	75.70%	20.00%	5.30%
Time(s)	19	266	3	1	1	1	1	1	1	1	2	2

TABLE III
PERFORMANCE OF DIFFERENT ACTIVE FUNCTIONS

Active fuctions	compet	elliotsig	hardlim	hardlims	logsig	netinv	poslin	purelin	radbas	radbasn	satlin	satlins	softmax	tansig	tribas
Testing accuracy	9.80%	90.40%	9.00%	10.30%	88.30%	20.00%	84.30%	90.60%	86.30%	89.00%	73.30%	88.70%	90.30%	92.70%	65.00%

In view of that these learning techniques are well-known model, so we just introduce the processes that we tune model parameters, and show some related performances briefly.

1) KNN: For KNN model, the value of K can affect the classification accuracy as small K results in great variance and large K brings huge bias in the results. As a result, we carefully tune K by varying it in the range of $[1, 50]$. As shown in Fig. 5, we finally select $K = 5$ as the optimal value due to its best performance.

2) SVM: For SVM model, we utilize one-versus-one technique [29], in which the kernel is the radial basis function (RBF) with parameters (C, γ), which are the penalty coefficient and kernel function coefficient, respectively. Fig. 6 infers that $(2^{10}, 2^{-10})$ can be chosen as the parameters of SVM to achieve a better result.

3) ANN: As shown in Fig. 3, each stroke has unique writing pattern. This means that we should not use too many layers and nodes to construct a deep neural network to differentiate them, as it cost too much time to training and tuning. For this reason, we only design a two-layer ANN with 10 nodes to achieve a balance between time cost and prediction performance. Results indicate that our ANN model also have a rather good accuracy. For ANN model, several key parameters and kernel functions, such as the number of layers (L), the number of nodes (N), training function (f) and activation function (ϕ) need to be carefully tuned in order to achieve satisfactory performance. As revealed in Tab. II and Tab. III, different training functions and active functions have a vital impact on prediction performance. Considering training efficiency, Levenberg-Marquardt algorithm was chosen since its low time complexity. Arbitrary value can be the input of tan-sigmoid function (ϕ_1), which has the best prediction accuracy, was used as the activation function of the first layer of our ANN model. Softmax function(ϕ_2) is able to output multiple classification probability, which performs well in results and thus are utilized in the second layer of ANN. Concluding above, we set these parameters as Tab. IV.

TABLE IV
PARAMETER SETTINGS OF ANN MODEL

Parameters	Value
Number of layers (L)	2
Number of nodes (N)	10
Training function (f)	Levenberg-Marquardt algorithm
Activation function (ϕ)	$\phi_1 = \frac{2}{1+e^{-2n}} - 1$ $\phi_2 = \frac{e^n}{\sum e^n}$

IV. EXPERIMENTS AND PERFORMANCE EVALUATION

In this section, we first describe the details of the experimental settings and implementation of our system. And then we conduct evaluation of our system from different aspects.

A. Experimental Setup and Data Collection

We implement AcouDigits on the Android platform with Samsung Galaxy Note 5. The acoustic signals are modulated by a sinusoidal scheme on 19 KHz frequency band, which is supported by most commercial devices. The reasons of choosing such high-frequency signals are two-fold. On one hand, acoustic signals in such high frequency range can sense motion with satisfactory resolution. On the other hand, signals

Fig. 7. Basic experiment scenarios.

received from this frequency range are usually inaudible for most of the human being. Moreover, the sampling rate is set to 44.1 KHz, which satisfies the Nyquist sampling theorem and is supported by current Android OS.

To do experiments, we recruit a total number of 10 participants with ages ranging from 21 to 31 (6 males and 4 females) who are students and staffs in our university. The experimental setup is shown in Fig. 7. As the participants have no knowledge of the experiments, we give a brief introduction of our project especially the experimental procedure. During experiments, we request them to write the basic digits (*i.e.*, $0-9$) in the natural way that they are used to as usual. In order to verify the robustness of AcouDigits, we conduct two major groups of experiments to collect data as follows.

- *Different participants.* Individuals are required to writing every digit for 200 times in total at about 8 cm from microphone. Thus, we finally obtain $20,000$ ($200\times10\times10$) instances. To make the experiments closer to practice, participants write each digit/letter every 5 times as a session and perform 4 sessions for each digit/letter alternately in a day. As a result, this part of experiments last for ten days.
- *Different distances.* To verify the robustness of AcouDigits, we also collect data at different positions of the writing finger and the device by varying their distance from 2 cm to 16 cm. Each participant is requested to write a single digit 50 times, and consequently a total of $24,000$ ($50\times10\times6\times8$) data sequences are collected. In this session, we only request 6 participants to take part in experiments in order to save time.

It is noted that the distance between a finger and the device as mentioned above is measured from its starting position and is not strictly restricted during writing. After data collection, we feed them into the data processing module as demonstrated in Section III. 80% data are utilized for training models and others are left for testing. In order to obtain the optimal performance of AcouDigits, we first implement the entire data processing module on a server, and obtain the required optimal parameters for this module. We then transform the entire system into the Android platform, and run an initial version of AcouDigits that takes the previously obtained optimal parameters into account.

B. Experimental Results

In this part, we evaluate the performance of AcouDigits from three main aspects, namely recognition performance, training overhead and user diversity.

1) Recognition Performance: Fig. 8 shows the overall performance of AcouDigits for KNN, SVM and ANN models. The corresponding experiments are conducted in the scenario where the distance between the finger and device is 8 cm. For each digit, the recognition accuracy is calculated by averaging the results over all participants. As shown in the figure, the overall recognition accuracy of SVM and ANN models are 89.5% and 91.7%, respectively, and are higher than that of KNN by 6.3% and 8.5%, respectively. Moreover, the running time of KNN model is much higher than that of SVM and ANN models. For each frame of signals, KNN model takes the entire feature sequence for the similarity calculation, *i.e.*, Dynamic Time Warpping (DTW) and the remaining noise may produce large bias in the results, and hence it incurs higher computational cost. Therefore, in the subsequent evaluation of AcouDigits, we only demonstrate the performance results obtained by SVM and ANN models. Fig. 9 shows the corresponding confusion matrix by averaging the performance of SVM and ANN models. Moreover, it is clear that different digits have different recognition accuracies due to the variance in writing process of certain digits.

2) Safe Distance: We have tested the performance of AcouDigits for different distances between the finger and the device, which verifies the robustness of the system. As shown in Fig. 10, the performance of AcouDigits decreases with the increasing distance due to the decay of echo intensity. However, within 8 cm, the performance remains relatively high with an accuracy no less than 91.5%. Even when the distance increases to 10 cm, AcouDigits still achieves an accuracy of 87.6%. Although such a distance is a not considerably large, we think it is suitable for a user to interact with a device especially wearables in practical usage scenarios.

3) Training Overhead: Training overhead is another important evaluation metric for a HCI system as this implies the quality of user experience. Consequently, we evaluate the performances of AcouDigits with the increasing number of training samples in Fig. 11. Clearly, the performance improves with the increasing number of training samples no matter it is SVM or ANN model. However, when the number of training samples exceeds the number around 40, the recognition accuracy increases much more slowly and remains nearly constant. Considering the trade-off between the accuracy and running time, we select 40 as a default number of samples to train SVM and ANN models.

4) Cross-person performance: We also conduct evaluation on cross-person performance, that is, training AcouDigits with one participant's data and testing it with another one's data. As there are 90 different training-testing pairs, we only display the results of five pairs that are randomly selected limited by the page space. The results are shown in Fig. 14. The average accuracies over different pairs for SVM and ANN are 75.4% and 78.0%, respectively. Compared with results in Fig. 8, the

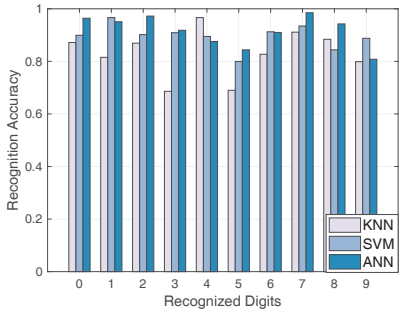


Fig. 8. The overall performance of AcouDigits for KNN, SVM and KNN models.

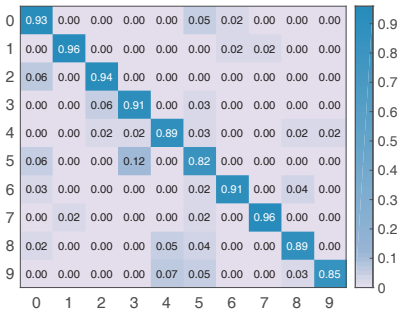


Fig. 9. The confusion matrix of AcouDigits while averaging the performance of SVM and ANN models.

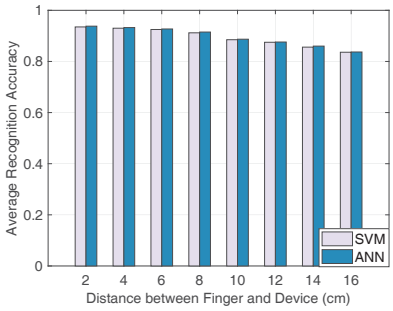


Fig. 10. The performance of AcouDigits for different distances between the finger and device.

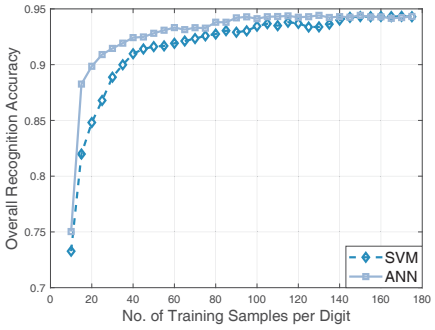


Fig. 11. The performance of AcouDigits for different numbers of training samples.

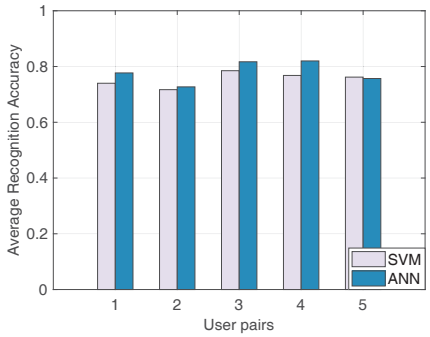


Fig. 12. The average accuracies of the selected five training-testing pairs.

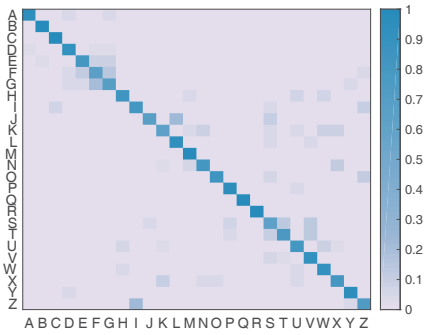


Fig. 13. The performance of AcouDigits in recognizing uppercase English letters.

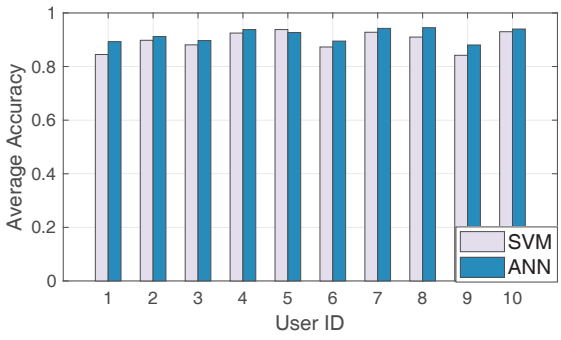


Fig. 14. The performance of AcouDigits for different participants in the experiments.

cross-person performance is much lower due to the differences in writing patterns of different participants. Nevertheless, we think that this does not affect the practicability much as people usually use their own devices like smartphones.

5) User Diversities: We also evaluate the performance of AcouDigits for different users considering the user diversities due to different writing habits. As mentioned above, we recruit six volunteers to take part in the experiments. We calculate the recognition accuracy for each participant, and the results are shown in Fig. 12. As we can see, the recognition accuracy of AcouDigits varies from (84.2%, 88.0%) to (94.8%, 95.2%) with (0.14%, 0.06%) variance among different participants due to different writing habits. However, from the variance of accuracy, we can claim that the performance of AcouDigits is consistent and satisfactory.

6) A Direct Extension to English Letters: Besides recognizing digits, we envision that AcouDigits can also be extended to recognize English letters. In order to verify this, we conduct experiments on a similar setup as is done for digits recognition. Specifically, we request six of the participants to write each English letter 100 times staying on the safe distance (*i.e.*, within 8 cm) of the device. Consequently, 15600 ($100\times6\times26$) signal consequences are collected. We take ANN as the learning model for the recognition purpose. Similarly, we feed 80% data to ANN models and others are left for testing. The resultant average recognition accuracy over all users are plot on a confusion matrix in Fig. 13. With the direct extension, the average accuracy in recognizing 26 letters is 87.4%, which is lower than that in recognizing digits. The main reason for this performance degradation is that several letters (*e.g.*, 'I' and 'Z', 'J' and 'L') have very similar writing forms. Therefore, these letters are recognized with low accuracy. However, we expect that with the further refinement of AcouDigits, the performance of recognizing letters will be improved. This is basically one of our future works.

V. Discussion

The robustness of AcouDigits: Since AcouDigits is built upon high-frequency acoustic signals, it is in nature resistant against noise interference by filtering out unwanted frequency components. Another interference is irrelevant motions except user's writing behavior. Similar to writing behavior, other irrelevant motions such as waving and walking shall have effect on signals as well. However, we notice that controlling emission power of acoustic signals shall effectively restrict the propagation range to a narrow range. Beyond this range, other motions nearly have little influence on the signals.

The future work of AcouDigits: In the present version of AcouDigits, we only consider the case where digital numbers are recognized. The principal future work is to extend AcouDigits as to recognize characters and then words and even sentences. The work flow of recognizing characters is much similar to this work. However, when extending to recognize words or sentences, linguistic models can be considered as an assistant tool to perform words predication or correction, so as to improve the recognition performance.

VI. Conclusion

Motivated by the increasing popularity of tiny wearable devices, we propose a touch-free interface, i.e., AcouDigits, using pervasive acoustic hardware embedded in present smart devices. AcouDigits enables a user to enter ten basic digits in the air using a finger and without wearing any additional device. With such an interface, a user can not only resolve the digits entry problem of small-size devices, such as giving a phone call, inputting bluetooth password, but also overcome the awkward situations like changing TV channel when clean hands are not available. We conduct extensive experiments to verify the effectiveness of AcouDigits. The results show that AcouDigits can recognize ten basic digits with high accuracy, and demonstrate the potential for future air-based text-entry interfaces using acoustic signals.

Acknowledgment

This research work was supported by the China NSFC Grant (61802264, 61872248, U1736207), Guangdong Natural Science Foundation 2017A030312008, Shenzhen Science and Technology Foundation (JCYJ20170302140946299, JCYJ20170412110753954), Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China (Grant No.161064), Guangdong Talent Project 2015TX01X111 and GDUPS (2015). This research work was also partially supported by Tianjin Key Laboratory of Advanced Networking (TANK), School of Computer Science and Technology, Tianjin University, Tianjin province, China, 300350. This research work was also partially supported by the project "PCL Future Regional Network Facilities for Large-scale Experiments and Applications (PCL2018KP001)", and Tencent Rhinoceros Birds- Scientific Research Foundation for Young Teachers of Shenzhen University. Kaishun Wu is the corresponding author.

References

[1] T. Page, "A forecast of the adoption of wearable technology," *International Journal of Technology Diffusion*, vol. 6, no. 2, pp. 12–29, 2015.
[2] L. Sun, S. Sen, D. Koutsonikolas, and K.-H. Kim, "Widraw: Enabling hands-free drawing in the air on commodity wifi devices," in *Proceedings of ACM MobiSys*, 2015.
[3] J. Wang, D. Vasisht, and D. Katabi, "RF-IDraw: virtual touch screen in the air using rf signals," in *Proceedings of ACM SIGCOMM*, 2014.
[4] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, "Tagoram: Real-time tracking of mobile rfid tags to high precision using cots devices," in *Proceedings of ACM MobiCom*, 2014.
[5] T. Wei and X. Zhang, "mtrack: High-precision passive tracking using millimeter wave radios," in *Proceedings of ACM Mobicom*, 2015.
[6] S. Agrawal, I. Constandache, S. Gaonkar, R. Roy Choudhury, K. Caves, and F. DeRuyter, "Using mobile phones to write in air," in *Proceedings of ACM MobiSys*, 2011.
[7] C. Amma, M. Georgi, and T. Schultz, "Airwriting: Hands-free mobile text input by spotting and continuous recognition of 3d-space handwriting with inertial sensors," in *Proceedings of IEEE ISWC*, 2012.
[8] M. Goel, L. Findlater, and J. Wobbrock, "Walktype: using accelerometer data to accomodate situational impairments in mobile touch screen text entry," in *Proceedings of ACM CHI*, 2012.
[9] T. Ni, D. Bowman, and C. North, "Airstroke: bringing unistroke text entry to freehand gesture interfaces," in *Proceedings of ACM CHI*, 2011.
[10] S. Nirjon, J. Gummeson, D. Gelb, and K.-H. Kim, "Typingring: A wearable ring platform for text input," in *Proceedings of ACM MobiSys*, 2015.
[11] K.-Y. Chen, D. Ashbrook, M. Goel, S.-H. Lee, and S. Patel, "AirLink: sharing files between multiple devices using in-air gestures," in *Proceedings of ACM UbiComp*, 2014.
[12] Y. Qifan, T. Hao, Z. Xuebing, L. Yin, and Z. Sanfeng, "Dolphin: Ultrasonic-based gesture recognition on smartphone platform," in *Proceedings of IEEE CSE*, 2014.
[13] S. Gupta, D. Morris, S. Patel, and D. Tan, "Soundwave: using the doppler effect to sense gestures," in *Proceedings of ACM CHI*, 2012.
[14] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "Fingerio: Using active sonar for fine-grained finger tracking," in *Proceedings of ACM CHI*, 2016.
[15] X. Yi, C. Yu, W. Xu, X. Bi, and Y. Shi, "Compass: Rotational keyboard on non-touch smartwatches," in *Proceedings of ACM CHI*, 2017.
[16] C. Yu, K. Sun, M. Zhong, X. Li, P. Zhao, and Y. Shi, "One-dimensional handwriting: Inputting letters and words on smart glasses," in *Proceedings of ACM CHI*, 2016.
[17] Apple Inc., "Siri APP," https://www.apple.com/ios/siri/, 2011, accessed on 2017-09-03.
[18] Microsoft Inc., "Cortana APP," https://www.microsoft.com/en-us/windows/cortana, 2014, accessed on 2017-09-03.
[19] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using wifi signals," in *Proceedings of ACM Mobicom*, 2015.
[20] M. T. I. Aumi, S. Gupta, M. Goel, E. Larson, and S. Patel, "DopLink: Using the doppler effect for multi-device interaction," in *Proceedings of ACM UbiComp*, 2013.
[21] S. Yun, Y.-C. Chen, and L. Qiu, "Turning a mobile device into a mouse in the air," in *Proceedings of ACM MobiSys*, 2015.
[22] Z. Sun, A. Purohit, R. Bose, and P. Zhang, "Spartacus: spatially-aware interaction for mobile devices through energy-efficient audio sensing," in *Proceedings of ACM MobiSys*, 2013.
[23] Z. Zhang, D. Chu, X. Chen, and T. Moscibroda, "SwordFight: Enabling a new class of phone-to-phone action games on commodity phones," in *Proceedings of ACM MobiSys*, 2012.
[24] W. Mao, J. He, and L. Qiu, "CAT: high-precision acoustic motion tracking," in *Proceedings of ACM Mobicom*, 2016.
[25] W. Wang, A. X. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," in *Proceedings of ACM Mobicom*, 2016.
[26] S. Yun, Y.-C. Chen, H. Zheng, L. Qiu, and W. Mao, "Strata: Fine-grained acoustic-based device-free tracking," in *Proceedings of ACM Mobisys*, 2017.
[27] S. W. Smith *et al.*, "The scientist and engineer's guide to digital signal processing," 1997.
[28] V. Madisetti, *The digital signal processing handbook*. CRC press, 1997.
[29] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE transactions on Neural Networks*, 2002.