VibWriter: Handwriting Recognition System based on Vibration Signal

Dian Ding, Lanqing Yang, Yi-Chao Chen, Guangtao Xue*,
Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
Email:{dingdian94, yanglanqing, yichao, gt_xue}@sjtu.edu.cn

***Abstract*—The efficiency of human-computer interaction is greatly hindered by the small size of the touchscreens on mobile devices, such as smart phones and watches. This has prompted widespread interest in handwriting recognition systems, which can be divided into active and passive systems. Active systems require additional hardware devices to perceive movements of handwriting or the tracking accuracy is not adequate for handwriting recognition. Passive methods use the acoustic signal of pen rubbing and are susceptible to environmental noise (above 60dB). This paper presents a novel handwriting recognition system based on vibration signals detected by the built-in accelerometer of smart phones. *VibWriter* is highly resistant to interference since the normal environmental noise will not cause the vibration of the accelerometer. Extensive experiments demonstrated the efficacy of the system in terms of accuracy in letter recognition (76.15%) and word recognition (88.14%) when dealing with words of various lengths written by various users in a variety of writing positions under a variety of environmental conditions.**

***Index Terms*—vibration signal, handwriting recognition**

I. Introduction

The shortcomings of touchscreen input methods have become increasingly obvious with the advent of smart phones, smart watches, and other intelligent devices [1]. Most of the researches on alternative input systems have focused on speech recognition [2] and handwriting recognition [3], [4]. Handwriting input is often the only option in cases where privacy is a concern.

Most existing handwriting recognition methods can be categorized as localization-based and scratch-based methods. Localization-based methods detect the movement of the user's hand or pen via inertial sensors [1] or wireless signals, such as acoustic signal [5], [6], [7], WiFi signal [3], and magnetic signal [8]. Methods based on WiFi signal [3] or magnetic signal [8] have limitations for experimental scenarios. The acoustic-based tracking methods [5], [6], [7] achieve millimetre-level tracking accuracy. Since the medium size of letters in handwriting is $2.5-3.5mm$ according the researches in graphology [9], these methods can still impair the recognition accuracy. Scratch-based methods [4], [10], [11] involve the detection of acoustic signals generated by dragging a pen or finger across a surface, but these methods are highly susceptible to environmental noise (above 60dB) [4], [10].

In this paper, we seek to overcome the shortcomings of existing handwriting recognition schemes by developing a system that uses the built-in accelerometer of the smart phone to detect the vibration signals generated by a pen writing on the desk. In experiments, *VibWriter* proves highly robust to interference from environment noise and vibrations. The system also demonstrates outstanding recognition performance under different conditions, such as different smart phones, different desks and different writing regions.

The development of *VibWriter* imposes some challenges:

(1) The sampling rate of the built-in accelerometer tends to be low and lacking in stability. This imposes daunting challenges in reconstructing and processing vibration signals from an input with limited bandwidth.

(2) The fact that the vibration signal indicating the start of a new letter is usually generated by a tap or swipe makes it difficult to differentiate between letters. Real-world writing scenarios also present numerous unexpected situations prompting the user to write more quickly or more slowly. Finally, a small time interval between letters can lead to signal overlap, whereas a large time interval can hinder signal separation.

(3) The removal of noise from the signal can be hindered by variations in noise characteristics over time.

VibWriter addresses these issues using the corresponding solutions listed below:

(1) Data missing from the vibration signal is reconstructed using the spline interpolation algorithm. The Xception module is used to extract deep features for the residual architecture and depth-wise separable convolution layers.

(2) A mean window is used to detect signal segments that are characteristic of handwriting. The problems of signal overlap and signal separation are dealt with by combining information in the time and frequency domains and selecting appropriate time for signal splitting and merging based on changes in signal strength.

(3) We develop a dynamic denoising algorithm, which uses the noise signal generated during idle periods as a reference.

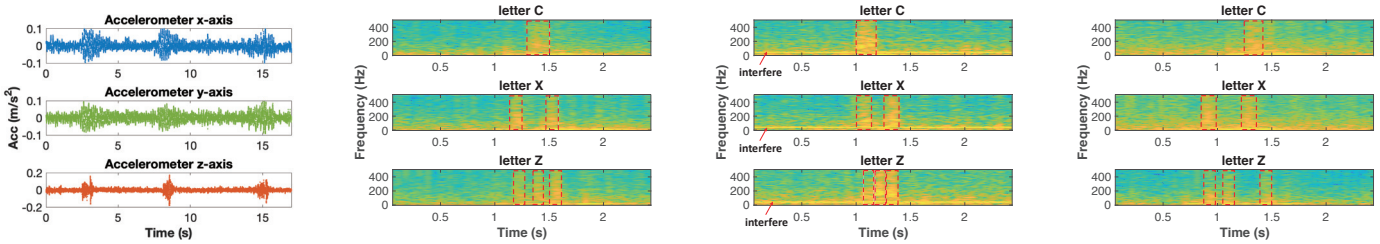
To the best of our knowledge, this is the first vibration-based handwriting recognition system. The main contributions are summarized as follows:

(1) We demonstrate that the built-in accelerometer of the smart phone provides the sensitivity and resolution required for the detection of vibration signals generated by handwriting.

(2) We develop the signal processing techniques required to deal with these vibration signals, including signal construction, feature extraction, and feature classification. We also resolve the problems of signal overlap and signal separation.

* Corresponding author.

978-1-6654-4108-7/21/$31.00



(a) Vibration signal of volunteer 1. (b) Frequency spectrum of volunteer 1. (c) Frequency spectrum of volunteer 1 with interferes. (d) Frequency spectra of volunteer 2.

Fig. 1. Preliminary experiments with Samsung S7 (495Hz): 1(a) and 1(b) Vibration signal and spectrum generated by writing the letters "C", "X" and "Z" of volunteer 1; 1(c) Spectrum of volunteer 1 with different interferes; 1(d) Spectrum generated by writing the letters of volunteer 2.

(3) We implement *VibWriter* on an Android smart phone. In experiments, the system achieves accuracy of 76.15% in letter recognition and 88.14% in word recognition.

II. Background

VibWriter uses the built-in accelerometer of a Samsung S7 to detect vibration signals generated by the desk when in contact with a pen. This section outlines preliminary experiments aimed at answering the following fundamental questions: i) Do the vibration signals generated by the desk produce characteristics of different letters? ii) Do the different environments and users affect the vibration signal?

In the first experiment, we seek to determine whether the vibration signals generated by the desk produce characteristics of different letters [12]. The accelerometer of smart phone can achieve the sampling rate of approximately $500Hz$ [13], and even a small strokes of $0.1s$ can generate 50 samples. Therefore, we try to recognize different handwriting letters with the vibration signal. One volunteer is tasked with writing the letters "C", "X", and "Z". As shown in Fig.1(a), the exceedingly weak amplitude of the vibration signals make it difficult to differentiate between the three letters directly. Besides, different letters comprise different numbers of strokes, as indicated by the spectrum in which the letter "Z" comprises three strokes, the letter "X" comprises two , and the letter "C" comprises only one stroke (see Fig.1(b)).

In the second experiment, we first test the vibration signals in different environments. When the volunteer is writing, we add different vibration disturbances such as arm movements and the fan. As shown in Fig.1(c), the vibration caused by the fan and the movements of the user's arm is concentrated in the lower frequency band (below $200Hz$), and the high frequency part of the vibration signal can still distinguish the strokes written by the volunteer. However, the uncertainty of vibration interference distribution puts forward the requirements for signal denoising.

Then, we invite another volunteer to write the letters as shown in Fig.1(d). We can also distinguish the strokes of the user from the spectrum. However, due to differences in pauses, stroke order and strength in the writing process, the differences in vibration signals make it difficult to popularize signal recognition.

Preliminary experiments prove that based on the vibration signal, the user's strokes can be recognized to distinguish handwriting in different environments. Nonetheless, it would be difficult to differentiate between all of the letters based solely on the number of strokes. When writing quickly, many letters would be indistinguishable from others with the same number of strokes (e.g., "D"and "P" or "C" and "O"). A feature extraction scheme is required for letter recognition.

III. System

As shown in Fig. 2, *VibWriter* comprises three modules: letter segmentation, letter recognition, and word suggestion. Vibration signal detected by the built-in accelerometer is first sent to the letter segmentation module to be divided into discrete segments. The letter recognition module identifies the different segments. Finally, the word suggestion module combines the letters into words. The three modules are described in detail below.

A. Letter Segmentation

As shown in Fig.1, our first objective is to detect the handwriting by amplitude variation. Unfortunately, real-world data acquisition can lead to a number of issues, such as unstable sampling rate, incomplete data segmentation, and letter concatenation. The proposed segmentation algorithm deals with these issues in two stages: interpolation and detection.

Interpolation: Obtaining the highest sampling rate from the built-in accelerometer precludes the stable sampling rate of raw data [13]. In most situations, more than half of the vibration signals are missing, such that the actual number of samples collected per second is roughly 490.

The accuracy of timestamps is $1ms$. Therefore, the ideal approach would involve upsampling the raw data to $1000Hz$. This linear interpolation approach has previously been used to stabilize the sampling rate [13]. However, when the time interval exceeds $4ms$, the complete cycle of the signal (above $250Hz$) is missing and cannot be recovered.

We compare a variety of interpolation algorithms [14], including spline interpolation, trigonometric interpolation and linear interpolation, as shown in Fig.4(a). Spline interpolation proves more effective than linear interpolation in the recovery of lost data over extended time intervals, and outperforms

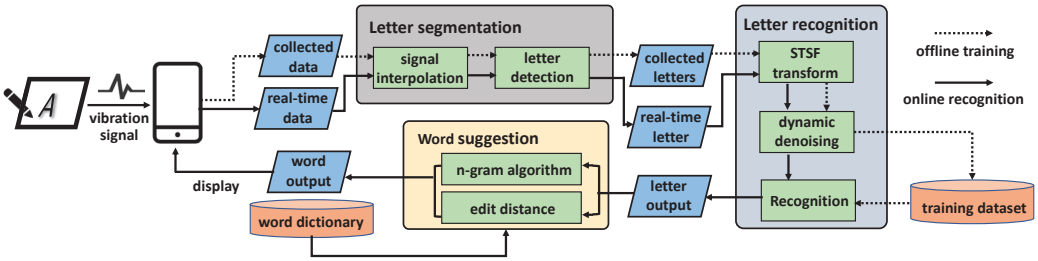


Fig. 2. Overview of *VibWriter*.

trigonometric interpolation in terms of how well the recovered signal fits the original data. Furthermore, the mean squared errors of the interpolation algorithms are 0.00468, 0.00585 and 0.00331 respectively.

Spline interpolation uses low-degree polynomials in each interval, and selects polynomial pieces in a manner that ensures a smooth fit when combined. For known points $(x_1, y_1), (x_2, y_2)$, the third-order polynomial is:

$$q(t) = (1-t(x))y_1+t(x)y_2+t(x)(1-t(x))((1-t(x))a+t(x)b) \quad (1)$$

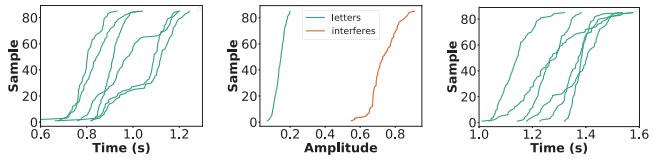
where

$$t(x) = \frac{x - x_1}{x_2 - x_1}$$
$$a = k_1(x_2 - x_1) - (y_2 - y_1)$$
$$b = -k_2(x_2 - x_1) + (y_2 - y_1)$$
$$k_1 = q^{'}(x_1)$$
$$k_2 = q^{'}(x_2)$$

Detection: Generally, the tap of a pen on the desk surface produces a distinctive vibration pattern indicating the beginning of writing. However, in some situations where the user seeks to write quietly, such as a meeting room, the writing process begins with a swipe. This situation makes it difficult to identify the start of writing. The signal produced by a tap presents an abrupt change in amplitude, whereas the amplitude of the signal produced by a swiping motion grows gradually. The common approach to segmentation often fails to identify vibration signals that begin with a swipe [4], [10]. We calculate the mean value of the vibration signal $S(t)$ with the sliding window $t_w = 100ms$.

Letter detection is based largely on three time thresholds T_1, T_2 and T_3, and three amplitude thresholds A_1, A_2 and A_3. T_1 and T_2 indicate the minimum and maximum lengths of the letters, whereas T_3 indicates the time interval between words. A_1 and A_2 indicate the maximum and minimum absolute values of $M(t)$, whereas A_3 indicates the minimum absolute value of interference. We use the time threshold to constrain the signal length of letters and words, and the amplitude threshold to judge the begin and end of the signal.

Peak selection is based on the amplitude threshold, where the start threshold is $M_{start} = 0.2 \times A_1 + 0.8 \times A_2$ and the end threshold is $M_{end} = 0.1 \times A_1 + 0.9 \times A_2$.



(a) Length of letters. (b) Amplitude of signals. (c) Length of intervals.

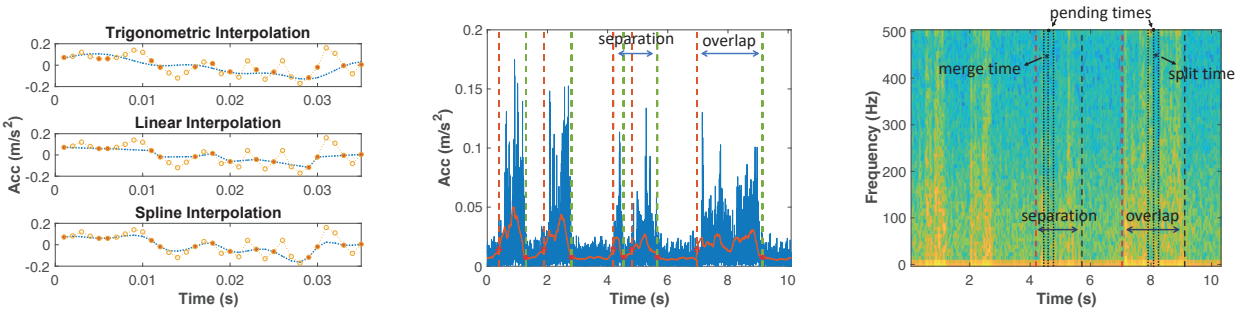
Fig. 3. Experiments on normal writing patterns in the time and amplitude domains: 3(a) Time elapsed while writing letters of different users; 3(b) Amplitudes of target signals and interference; 3(c) Intervals between words of different users.

In instances where the amplitude of $M(t_0)$ exceeds M_{start}, timestamp t_0 indicates the start of a writing segment. As long as the user is writing in a normal manner, it is possible to identify the end of a writing segment based on M_{end}, as shown in Fig.4(b).

As shown in Fig.3(a), preliminary experiments show that the handwriting time remains stable for most users. Therefore, under normal circumstances, it can be assumed that the users write in the block-letter style. However, we observe a number of special situations in which the signal is difficult to segment. In cases where the time interval between letters is short, the vibration signals of different letters can overlap in the time domain, due to the vibration signal lingering for a few milliseconds after writing ceases. Signal separation can also be hindered when the writing process is interrupted and will cause the incomplete segmentation. Besides, there are vibration interferes such as finger tapping on the desk, which can also affect the signal detection.

First, We set $t_{segment}$ as the length of the segment. If $t_{segment} > T_2$, the segment is identified as a combination of two letter signals. T_2 represents the maximum length of a single letter according to our experiment in Fig.3(a). We can locate a candidate split location, based on $Min\{M(t)\}$ in the time domain. As shown in Fig.1, the high frequency components of the vibration signal are mainly concentrated at the beginning of the signal. Combined with changes in signal strength in the spectrum, we can define the point with the weakest signal strength as the split point, as shown in Fig.4(c).

If $t_{segment} < T_1$, then it is designated a stroke of a letter. T_1 represents the minimum length of a single letter in Fig.3(a). Due to the remaining effect, the simple stitching of two segments is not good choice. Based on the observation



(a) Interpolations for missing data. (b) Detection of letters. (c) Detection of letters.

Fig. 4. Letter segmentation metrics: 4(a) Results of various interpolation methods: the red dots are the vibration signals (hollow dots are stable samples, solid dots are unstable samples), the blue lines are the interpolation results; 4(b) Vibration signal (blue) and corresponding weighted mean signal (red) showing different writing conditions: normal fast writing, an interruption during fast writing (with an interval during one letter) and continuous writing (without interval between letters). The dotted line indicates the results of segmentation based on amplitude; 4(c) Proposed solution to deal with signal separation and overlap caused by interruption and continuous writing.

of the spectrum above. We can define the point with the weakest signal strength as the merge point, so as to remove the remaining effect of the segment, as shown in Fig.4(c).

Then, we set $a_{segment}$ as the maximum amplitude of the segment. If $a_{segment} > A_3$, then it is designated the vibration interfere. A_3 represents the distinguishing threshold between handwriting signal and vibration interference. Since the amplitude of segments differ considerably from the interference according to preliminary experiments, as shown in Fig.3(b). In addition to large vibration disturbances such as knocking on the desk, minor disturbances such as common fans and keyboards on the desk will also affect the system. We further analyze these interferences in Section.IV-C.

Finally, as shown in Fig.3(c), the length of intervals between words tends to be uniform under normal writing conditions. Thus, intervals exceeding T_3 are designated as the end of a word, and T_3 represents the distinguishing threshold between letters and words.

B. Letter Recognition

Preprocessing: We adopt Short-time Fourier Transform (STFT) to generate features in the frequency domain. The vibration signals of the three axes are converted into a STFT matrix representing the magnitude and phase of each frame and frequency, as follows:

$$STFT\{x[t]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{+\infty} x[n]\omega[n-m]e^{-j\omega n} \tag{2}$$

where ω represents the frequency of window function, and m represents the scale of window function.

The sampling rate of the built-in accelerometer ($1kHz$) is far lower than the acoustic signal of handwriting [11], [10], [4], [15], and the spectral distribution of signals and noise is similar. As shown in Fig. 1(d), the amplitude of noise signals below $100Hz$ far exceeds that of higher frequency signals. Furthermore, signals associated with ambient noise do not remain stable throughout the writing process. Thus, noise removal should be a dynamic process implemented only at specific time points. We develop a dynamic denoising algorithm, which identifies noise based on a reference signal collected during idle periods. We begin by establishing a noise sample $\hat{S}_{noise} = [s_1, s_2, ..., s_l]$, and then update the sample as:

$$\hat{S}_{noise} = \frac{1}{N}\sum_{i=1}^{N} S_{noise_i} \tag{3}$$

where l indicates the length of the noise sample according to different handwriting segments. S_{noise} preserves the noise signal between letters and words, and N represents the number of samples in S_{noise}. Then, we can denoise the signal with the spectrum subtraction [16]:

$$\|Y(k)\|^2 = \|S_{signal}(k)\|^2 - \|\hat{S}_{noise}(k)\|^2 \tag{4}$$

where k represents the frequency range of the signal, $S_{signal}(k)$ and $\hat{S}_{noise}(k)$ represent the handwriting sample and the noise sample respectively. For each signal, we use the latest noise signal to update the noise sample.

Classification: Convolutional neural network (CNN) have proven highly effective in spectrum classification [4], [10]. The spectral width of vibration signals is far narrower than acoustic signals. Therefore, the module have to extract handwriting features at various scales, (e.g., single taps, single strokes, and entire letters). As shown in Fig.5, the Xception model [17] takes advantages of ResNet [18] and Inception [19]. As the model gets deeper, problems such as gradient disappearance arise. The residual structure in Xception (the arcs in the Basic Block) effectively solves this problem and enables features of different depths in the model to be fused. Second, the deepening of the model inevitably increases the computational burden on the hardware. the Xception model uses an Separable Convolution (the yellow layer in the Basic Block), which splits the normal convolution into two parts: Channel-wise Convolution and Point-wise Convolution. Channel-wise Convolution extracts features separately for individual channels in the feature, and Point-wise Convolution aggregates the feature points in different channels by 1×1 convolution. Thus, the

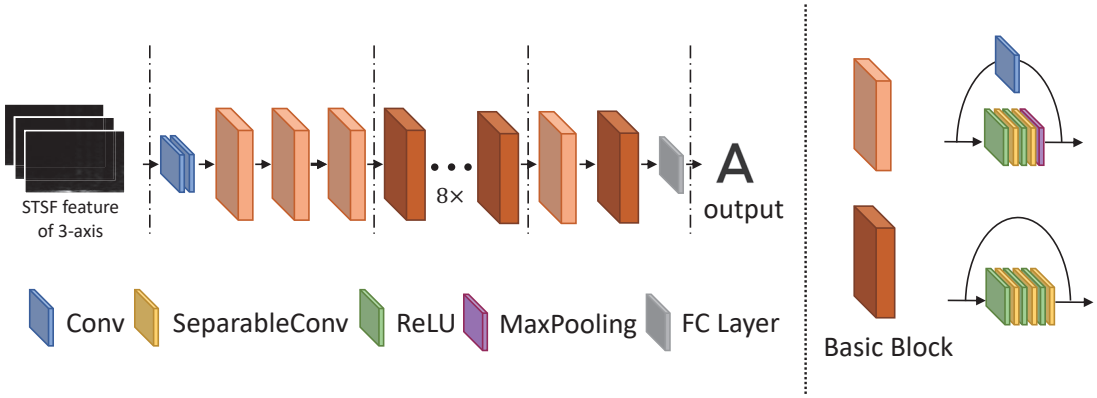


Fig. 5. Architecture of the Xception model.

$n \times n \times m$ parameters (m represents the number of channels) required for ordinary convolution are reduced to $n \times n + m$.

To further improve the accuracy of the model, we employ Focal Loss to facilitate learning using difficult samples [20]:

$$FL(p_t) = -\alpha(1-p_t)^{\gamma} log(p_t) \tag{5}$$

where p_t represents the output of the model, α and γ are correlation coefficients. $\alpha(1-p_t)^{\gamma}$ reverses with the difficulty of sample, so as to strengthen the difficult samples.

C. Word Suggestion

We notice the fact that users often write a word rather than a single letter. Therefore, we develop a word suggestion algorithm to enhance handwriting recognition performance at the word level.

N-gram algorithm Language models are widely used in natural language processing (NLP) [21]. We employ the N-gram to determine the probability distribution of letters in words. The chain rule of letters is defined as follows:

$$P(\omega_1, \omega_2, ..., \omega_n) = P(\omega_1)P(\omega_2|\omega_1)\cdots P(\omega_n|\omega_1, ..., \omega_{n-1}) \tag{6}$$

where $\omega_i, i \in [1, n]$ represents the letter in the word. The conditional probability of each letter occurrence is calculated in terms of maximum likelihood, as follows:

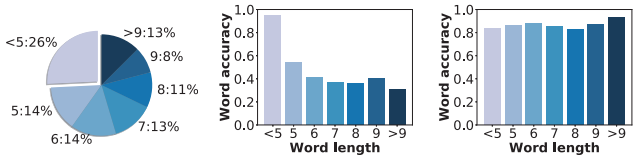
$$P(\omega_i|\omega_1, ..., \omega_{i-1}) = \frac{C(\omega_1, \omega_2..., \omega_i)}{\sum_{\omega} C(\omega_1, \omega_2..., \omega_i, \omega)} \tag{7}$$

where $C(\cdot)$ represents the number of times a string appears in the dataset. Obviously, it would be unrealistic to directly calculate $P(\omega_i|\omega_1, ..., \omega_{i-1})$ based directly on maximum likelihood estimation. Assuming that the probability of current letter occurring depends only on the the first $n-1$ letters, we obtain the following result:

$$P(\omega_i|\omega_1, ..., \omega_{i-1}) = P(\omega_i|\omega_{i-n+1}, ..., \omega_{i-1}) \tag{8}$$

Based on the above formula, the 3-gram language model is defined as follows:

$$P(\omega_i|\omega_1, ..., \omega_n) = \prod_{i=1}^{n} P(\omega_i|\omega_{i-1}, \omega_{i-2}) \tag{9}$$



(a) Word distribution. (b) N-gram algorithm. (c) Edit distance algorithm.

Fig. 6. Word suggestion results: 6(a) Distribution of words of various lengths among the 5000 most common words in COCA; 6(b) and 6(c) Accuracy in word identification respectively using N-gram and Edit Distance algorithms.

Edit distance It can be noted that accuracy in correcting misspelled words is closely related to the lengths of the words. As shown in Fig.6(b), when the length exceeds five letters, the accuracy of word suggestion schemes decreases significantly. Thus, we analysis the length distribution of the 5000 most commonly used words in the Corpus of Contemporary American English (COCA) in Fig.6(a). The words exceeding 6 letters make up more than half of the total; therefore, we focus on longer words using the edit distance algorithm.

Edit distance refers to the minimum number of editing operations required to change from one string to another. Permitted editing operations include replacing one character with another, inserting one character, and deleting one character. The shortest edit distance between the first i characters of string a and the first j characters of string b can be written as $Lev_{a,b}(i,j)$. The recursive formula used to determine the edit distance between two strings is as follows:

$$Lev_{a,b}(i,j) = \begin{cases} max(i,j) & if min(i,j) = 0 \\ min \begin{bmatrix} Lev_{a,b}(i-1,j)+1 \\ Lev_{a,b}(i,j-1)+1 \\ Lev_{a,b}(i-1,j-1) \end{bmatrix} & a_i = b_j \\ min \begin{bmatrix} Lev_{a,b}(i-1,j)+1 \\ Lev_{a,b}(i,j-1)+1 \\ Lev_{a,b}(i-1,j-1)+1 \end{bmatrix} & a_i \neq b_j \end{cases} \tag{10}$$

As shown in Fig.6(c), the edit distance greatly improve accuracy in correcting spelling errors in long words. Thus,

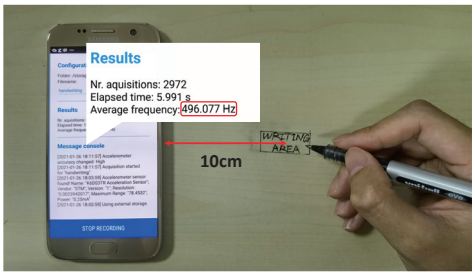


Fig. 7. Experimental Setup.

we employ the N-gram algorithm for words of less than five letters and edit distance for longer words.

IV. Evaluation

A. Experimental Setup

Hardware. *VibWriter* is implemented on a Samsung S7 and a MacBook Pro (Intel Core i9 CPU@2.3GHz and 16GB RAM) is implemented as the server. Based on the built-in accelerometer[1], we can achieve a sampling rate of about $490Hz$ [22], [13]. We conduct our experiments in the normal laboratory. As shown in Fig.11, we collect the training set and test set on a wooden desk. The smart phone is placed in the centre of the desk, perpendicular to the lower edge of the desk. The writing region is $10cm$ to the right of the smart phone.

Training set. We first invite six volunteers to write samples of 26 uppercase letters with a gel pen as a training set. Two of the volunteers write the 26 letters 60 times each, whereas the rest of the volunteers write the letters 20 times each. All the volunteers write directly on the desk at their own speeds, strength and in any order they wished.

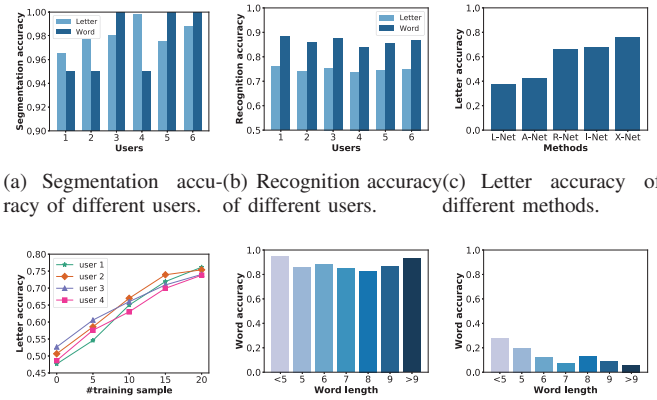
Test set. The volunteers are then tasked with writing the top 20 words of each length in COCA to test the overall accuracy of the system.

Parameter. For segmentation, we set the minimum and maximum length of letters $T_1 = 0.4s$, $T_2 = 1.5s$, minimum time of the word interval $T_3 = 1s$ and the minimum absolute value of interferes $A_3 = 0.4$ according to our experimental observation in Fig.3. We introduce the parameters in details in Section.III-A. For letter recognition, we set the segment and the overlap of STFT at 128 and 120. For Xception, we set the batch size at 32 for 40 epochs. We also use the Adam algorithm with a learning rate of 0.0008. Finally, we set the Focal Loss coefficients $\alpha = 0.2$ and $\gamma = 3$ [20].

B. Micro Benchmarks

In this section, we evaluate the performance of three main components of *VibWriter*. For each volunteer, we build a handwriting recognition model. For the volunteers who write letters for 60 times, we can build the model with their own handwriting samples. For the rest volunteers, we use the handwriting samples of two volunteers to build the basic model, and then fine-tune the model with their own samples.

[1]We use a third-party application AccDataRec for diaplay.



(a) Segmentation accuracy of different users. (b) Recognition accuracy of different users. (c) Letter accuracy of different methods.

(d) The effect of #training sample. (e) Accuracy with word suggestion. (f) Accuracy without word suggestion.

Fig. 8. Accuracy of the VibWriter system.

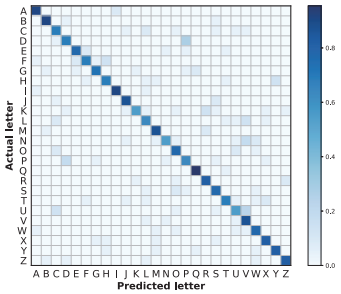


Fig. 9. Confusion matrix of letter recognition.

1) Letter Segmentation: First, we evaluate the accuracy of the system in terms of letter segmentation, as shown in Fig.8(a). The segmentation algorithm outlined in Section.III-A prove highly effective in dealing with signal overlap and signal separation. However, there are some cases that fluctuations in the vibration signals are too weak to detect. Those situations are deemed segmentation failures. The average accuracy results in the segmentation of letters and words were 98.07% and 97.5%, respectively. Overall, this degree of accuracy should suffice for most practical applications.

2) Letter Recognition: We use the top-1 output of the network as the recognition result. As shown in Fig.9, the average accuracy in letter recognition is 75.69%. Analysis of misclassification reveals that around 20% of the letters "K" and "N" are misidentified as "R" and "V", respectively. Clearly, a word suggestion algorithm is required to achieve reasonable recognition performance.

Xception is compared with other classification methods, including LeNet, AlexNet, ResNet and Inception. As shown in Fig.8(c), the accuracy of Xception is far higher than that of the other classification algorithms. We reduce the number of training sets via model fine-tuning, as shown in Fig.8(d).

3) Word Suggestion: The performance of the VibWriter system using the N-gram algorithm for short words and the edit distance algorithms for longer words is verified by counting the number of correct words suggestions. As shown in Fig.8(e), the proposed algorithms achieve overall accuracy of

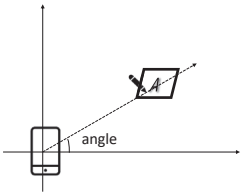


Fig. 10. Angle of the smart phone.

88.14% for words of various lengths. The inter-user accuracy of the system is shown in Fig.8(b). As shown in Fig.8(f), without the word suggestion algorithms, the average overall accuracy in word recognition is only 13.57%.

C. Macro Benchmarks

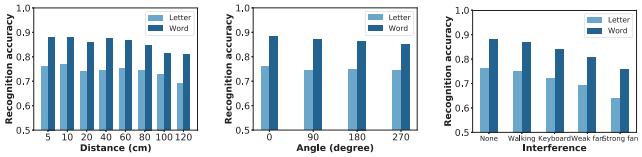
In this section, we evaluate the performance of *VibWriter* under a variety of conditions. In each experiment, we vary only one variable, such as the writing distance, writing angle, writing angle, etc.

1) Writing Location: The distance between the smart phone and the handwriting region is experimented by moving the phone in a horizontal direction, in a range of $5cm$ to $120cm$ from the handwriting region. During this process, the angle of the phone is not changed. Then, the phone is then placed back in its original position ($10cm$ to the left of the handwriting region) and the angle of the phone is changed, as shown in Fig.10. The volunteers are tasked with writing the same words as test set. Overall, *VibWriter* achieves high accuracy in terms of handwriting recognition regardless of the distance and angles between the writing position and the smart phone, as shown in Fig.11(a) and Fig.11(b).

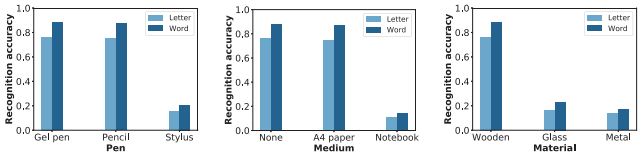
2) Vibration Interference: Unlike the interference discussed in Section.III-A, the disturbances of minor vibrations could potentially interfere with *VibWriter*, such as the vibration of the desktop fan, people walking around, tapping on the keyboard, etc. We evaluate each of these Interference separately. The desktop fan is placed on the desk at a distance of $5cm$ from directly above the smart phone to simulate interference from electronic devices. Besides, two volunteers are asked to walk around the desk or tap the keyboard on the desk to simulate the other two interferences. The keyboard is placed $20cm$ to the right of the writing region. Then, the other volunteers are tasked with writing the same words as test set.

As shown in Fig.11(c), vibration interference have an impact on the vibration signal. Nonetheless, the dynamic denoising algorithm (described in Section. III-B) is able to maintain word recognition accuracy above 75%. Clearly, *VibWriter* is robust to most of the vibration-related interference commonly encountered in the real environments.

3) Different vibration sources: The vibration signal generated by writing is closely related to the vibration source, such as different pens (include gel pen, pencil and stylus), different medium (include a piece of A4 paper and notebook) and desk material (include wooden, glass and metal). The volunteers are tasked with writing the same word as the test set under different conditions. We verified different pens, medium and



(a) Influence of different distances. (b) Influence of different angles. (c) Influence of different vibration interferences.



(d) Influence of different pens. (e) Influence of different medium. (f) Influence of different materials.

Fig. 11. Evaluation of VibWriter under different conditions.

desk materials separately. We also kept the phone position, writing distance and other conditions constant.

As shown in Fig.11(d), the accuracy of the stylus is significantly lower than that of hard pens, because the vibration signal generated by the softer tip is weak. Therefore, we do not recommend writing with stylus.

Fig.11(e) gives the results of different medium, the results show that notebook has worst accuracy of 14.16%. Since the medium between the pen tip and the desk will seriously affect the propagation of the vibration signal, especially when the contact between the medium and the desk is loose or spaced, the vibration signal may be completely isolated.

The different desk materials also affect the recognition accuracy, as shown in Fig.11(f). Wooden desks are usually rough, whereas glass and metal desks are smoother and produce vibration signals of lower amplitude than wooden desks.

D. System Evaluation

1) Responsiveness: Latency (delays in system response) is a crucial issue in real-time input systems. In assessing the responsiveness of the overall system, we measure the time that elapsed between receiving a signal and outputting a result. The average latency in recognizing 520 letters is $165ms$. The average latency in recognizing 120 words is $239ms$. These results indicate that the responsiveness of *VibWriter* is sufficient for real-time operations.

2) User Study: A survey is conducted to collect feedback from the volunteers in terms of accuracy, input speed, responsiveness, and security. Scores are based on satisfaction with 5 points ranging from very unsatisfied (1) to very satisfied (5).

As shown in Tab.I, more than 85% of the volunteers express satisfaction with the system in terms of accuracy, input speed, and system security, whereas 75% are satisfied with the responsiveness of the system. Some of the volunteers comment that *VibWriter* is less susceptible to eavesdropping than conventional touchscreen input methods.

TABLE I
USER SATISFACTION OF *VibWriter*.

Satisfaction	Accuracy	Speed	Delay	Security
Very Satisfied	8	7	5	10
Satisfied	9	11	10	8
Normal	3	2	5	2
Unsatisfied	0	0	0	0
Very Unsatisfied	0	0	0	0

V. RELATED WORK

A. Localization-based Methods

The main idea of localization-based method is to recover the user's writing trajectory by tracking hand or pen in the space during the writing process. The major approaches ever used are motion-based and wireless signal-based.

Motion-based methods. These methods usually need to adopt embedded devices with built-in sensors such as gyroscope and accelerometer. [23] utilized the gyroscope and accelerometer built in the smart watch to track the movement of the user's hand. GyroPen [24] treated smart phones as pens, and the built-in sensors are used to track the user's actions and recognize the handwriting letters. Pentelligence [1] integrated the microphone and accelerometer into an electronic pen, combining the sound of writing with the moving information of the pen to recognize the user's handwriting.

Wireless signal-based methods. Wireless signal-based methods use wireless signals to sense the movements of the user's hand or pen, such as light, Wi-Fi and magnetic signal. WiReader [3] used Wi-Fi signal to sense the movement of user's hand based on Channel State Information. MagHacker [8] used the magnetic sensor built into smart phones to detect changes in the magnetic field of stylus during the writing process. Acoustic-based tracking methods [5]–[7], [25], [26] achieved millimetre-level tracking accuracy, the tracking error can increase as the writing distance increases. [7] showed that the error increases from $5mm$ to $15mm$ while the distance increases from $10cm$ to $40cm$. According to the researches in graphology [9], the medium size of handwriting letter is $2.5 - 3.5mm$. Therefore, these methods can still impair the recognition accuracy [27], [28]. As a comparison, *VibWriter* uses the built-in accelerometer of the smart phone. During the evaluation, the size of handwriting letters is around $5mm$ and the recognition accuracy remains similar across distances from $10cm$ to $60cm$.

B. Scratch-based Methods

Scratch-based handwriting methods use the acoustic signal caused by the friction during handwriting process. WordRecorder [10] used the spectrum diagram of the acoustic signals of single letter. WritingRecorder [4] designed the Inception-LSTM module to extract deep local features and time-series relations between frames. Ipanel [11] found that the acoustic signals caused by finger sliding against the desk depend on different movements. However, the scratch-based methods are sensitive to ambient noise, and the recognition accuracy decreases significantly when the noise is above $60dB$. Specifically, WordRecorder [10] showed the letter recognition accuracy is reduced by 37% from 79.8% to 50% with 60dB noise (while that of *VibWritter* is 76.2%); WritingRecorder [4] showed the word recognition accuracy is reduced by 19.8% from 92.8% to 74.4% with 65dB noise (while that of *VibWritter* is 88.1%). On the other hand, *VibWriter* is robust against both environmental sound noise and vibration noise.

C. Vibration-based application

Vibration signals are closely related to daily behaviors, such as walking [29], talking [13], [22] and authentication [30]–[32]. FootprintID [29], [33] used the vibration signal of the floor when walking to identify different users. Spearphone [22] and paper [13] used the effect of the phone's built-in speaker on the built-in accelerometer to steal the acoustic signal through the vibration signal. SurfaceVibe [12] proposed a vibration-based interaction tracking system for multiple surface types. [30], [31] enabled user authentication by means of user characteristics sensed by vibration signals.

VI. DISCUSSION AND CONCLUSION

Implement on smart watches. Due to hardware limitations, the sampling rate of accelerometers in smartwatches is approximately $100Hz$. Coarse-grained data cannot be used to identify the user's writing. While we believe that as smartwatches continue to be updated, *VibWriter* can be applied to smart watches and other mobile devices.

Sampling rate of smart phones. Taking an Android phone as an example, setting the highest sample rate $(SENSOR\_DELAY\_FASTEST)$ [13] will suffer from the problem of unstable sampling rate. The second highest sampling rate $(SENSOR\_DELAY\_GAME)$ has a delay of $20ms$, and the accelerometer has a sampling rate of $50Hz$ and a bandwidth of $25Hz$, which cannot be used to recognize the handwriting letters.

This paper introduces a novel handwriting recognition system based on vibration signals. The proposed VibWriter system is able to overcome instabilities in sampling rates and does not require external hardware devices. Extensive experiments demonstrated the efficacy of the system in terms of accuracy in letter detection (76.15%) and word detection (88.14%) when dealing with words of various lengths written by various users in a variety of positions under a variety of environment conditions.

In future work, we will extend the system to include lowercase letters and numbers, and develop a recognition system that runs entirely on the smart phone. Additional methods will be included to improve the recognition accuracy, including sentence-based suggestion and the fusion of vibration signals with other sensors, such as acoustic and gyroscope signals.

ACKNOWLEDGMENT

This work is supported by NSFC (61936015, U1736207, 62072306), Startup Fund for Youngman Research at SJTU, and Program of Shanghai Academic Research Leader (20XD1402100).

REFERENCES

[1] M. Schrapel, M.-L. Stadler, and M. Rohs, "Pentelligence: Combining pen tip motion and writing sounds for handwritten digit recognition," 04 2018, pp. 1–11.

[2] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques," *CoRR*, vol. abs/1003.4083, 2010. [Online]. Available: http://arxiv.org/abs/1003.4083

[3] Z. Guo, F. Xiao, B. Sheng, H. Fei, and S. Yu, "Wireader: Adaptive air handwriting recognition based on commercial wi-fi signal," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[4] H. Yin, A. Zhou, G. Su, B. Chen, L. Liu, and H. Ma, "Learning to recognize handwriting input with acoustic features," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 2, Jun. 2020. [Online]. Available: https://doi.org/10.1145/3397334

[5] K. Wu, Q. Yang, B. Yuan, Y. Zou, R. Ruby, and M. Li, "Echowrite: An acoustic-based finger input system without training," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1789–1803, 2021.

[6] S. Yun, Y.-C. Chen, H. Zheng, L. Qiu, and W. Mao, "Strata: Fine-grained acoustic-based device-free tracking," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 15–28. [Online]. Available: https://doi.org/10.1145/3081333.3081356

[7] W. Wang, A. X. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 82–94. [Online]. Available: https://doi.org/10.1145/2973750.2973764

[8] Y. Liu, K. Huang, X. Song, B. Yang, and W. Gao, "Maghacker: Eavesdropping on stylus pen writing via magnetic sensing from commodity mobile devices," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 148–160.

[9] Study of Handwriting: Size of Letters in Handwriting. [Online]. Available: https://www.handwriting-graphology.com/study-of-handwriting/

[10] H. Du, P. Li, H. Zhou, W. Gong, G. Luo, and P. Yang, "Wordrecorder: Accurate acoustic-based handwriting recognition using deep learning," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 1448–1456.

[11] 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," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 3, no. 1, Mar. 2019. [Online]. Available: https://doi.org/10.1145/3314390

[12] S. Pan, C. G. Ramirez, M. Mirshekari, J. Fagert, A. J. Chung, C. C. Hu, J. P. Shen, H. Y. Noh, and P. Zhang, "Surfacevibe: Vibration-based tap swipe tracking on ubiquitous surfaces," in *2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2017, pp. 197–208.

[13] Z. Ba, T. Zheng, X. Zhang, Z. Qin, B. Li, X. Liu, and K. Ren, "Learning-based practical smartphone eavesdropping with built-in accelerometer," 01 2020.

[14] P. Getreuer, "Linear Methods for Image Interpolation," *Image Processing On Line*, vol. 1, pp. 238–259, 2011.

[15] M. Zhang, P. Yang, C. Tian, L. Shi, S. Tang, and F. Xiao, "Soundwrite: Text input on surfaces through mobile acoustic sensing," in *Proceedings of the 1st International Workshop on Experiences with the Design and Implementation of Smart Objects*, ser. SmartObjects '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 13–17. [Online]. Available: https://doi.org/10.1145/2797044.2797045

[16] A. S. Rathore, W. Zhu, A. Daiyan, C. Xu, K. Wang, F. Lin, K. Ren, and W. Xu, "Sonicprint: A generally adoptable and secure fingerprint biometrics in smart devices," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 121–134.

[17] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[21] P. Nather, "N-gram based text categorization," 2005.

[22] S. A. Anand, C. Wang, J. Liu, N. Saxena, and Y. Chen, "Spearphone: A speech privacy exploit via accelerometer-sensed reverberations from smartphone loudspeakers," *CoRR*, vol. abs/1907.05972, 2019. [Online]. Available: http://arxiv.org/abs/1907.05972

[23] H. Jiang, "Motion eavesdropper: Smartwatch-based handwriting recognition using deep learning," in *2019 International Conference on Multimodal Interaction*, ser. ICMI '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 145–153.

[24] T. Deselaers, D. Keysers, J. Hosang, and H. A. Rowley, "Gyropen: Gyroscopes for pen-input with mobile phones," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 2, pp. 263–271, 2015.

[25] S. Yun, Y.-C. Chen, and L. Qiu, "Turning a mobile device into a mouse in the air," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 15–29.

[26] W. Mao, M. Wang, W. Sun, L. Qiu, S. Pradhan, and Y.-C. Chen, "Rnn-based room scale hand motion tracking," in *The 25th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '19. New York, NY, USA: Association for Computing Machinery, 2019.

[27] H. Pan, Y.-C. Chen, Q. Ye, and G. Xue, "Magicinput: Training-free multi-lingual finger input system using data augmentation based on mnists," in *IPSN'21*.

[28] Y. Zhang, W.-H. Huang, C.-Y. Yang, W.-P. Wang, Y.-C. Chen, C.-W. You, D.-Y. Huang, G. Xue, and J. Yu, "Endophasia: Utilizing acoustic-based imaging for issuing contact-free silent speech commands," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 1, Mar. 2020.

[29] S. Pan, N. Wang, Y. Qian, I. Velibeyoglu, H. Y. Noh, and P. Zhang, "Indoor person identification through footstep induced structural vibration," in *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 81–86. [Online]. Available: https://doi.org/10.1145/2699343.2699364

[30] X. Xu, J. Yu, Y. chen, Q. Hua, Y. Zhu, Y.-C. Chen, and M. Li, "Touchpass: Towards behavior-irrelevant on-touch user authentication on smartphones leveraging vibrations," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3372224.3380901

[31] J. Liu, C. Wang, Y. Chen, and N. Saxena, "Vibwrite: Towards finger-input authentication on ubiquitous surfaces via physical vibration," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 73–87. [Online]. Available: https://doi.org/10.1145/3133956.3133964

[32] C.-W. You, Y. Chuang, H.-Y. Lin, J.-T. Tsai, Y.-C. Huang, C.-H. Kuo, M.-C. Huang, S. J. Wu, F. W. Liu, J. Y.-J. Hsu, and H.-C. Wu, "Sobercomm: Using mobile phones to facilitate inter-family communication with alcohol-dependent patients," vol. 3, no. 3, 2019.

[33] S. Pan, T. Yu, M. Mirshekari, J. Fagert, A. Bonde, O. J. Mengshoel, H. Y. Noh, and P. Zhang, "Footprintid: Indoor pedestrian identification through ambient structural vibration sensing," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 3, Sep. 2017. [Online]. Available: https://doi.org/10.1145/3130954