FaceInput: A Hand-Free and Secure Text Entry System through Facial Vibration

Maoning Guan, Wenqiang Chen, Yandao Huang, Rukhsana Ruby and Kaishun Wu

July 18, 2019

FaceInput: A Hand-Free and Secure Text Entry System through Facial Vibration

Maoning Guan*, Wenqiang Chen*, Yandao Huang*, Rukhsana Ruby*, Kaishun Wu*,†
*College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China
†PCL Research Center of Networks and Communications, Peng Cheng Laboratory, Shenzhen, China
{guanmaoning2018,chenwenqiang2016,huangyandao}@email.szu.edu.cn, {ruby,wu}@szu.edu.cn

***Abstract*—Wearable wristbands have become prevailing in the recent days because of their small and portable property. However, the limited size of the touch screen causes the problems of fat fingers and screen occlusion. Furthermore, it is not available for users whose hands are fully occupied with other tasks. To break this bottleneck, we propose a portable, hand-free and secure text-entry system, called *FaceInput*, which firstly uses a single small form factor sensor to accomplish a practical user input via facial vibrations. To sense the tiny facial vibration signals, we design and implement a double-stage amplifier whose maximum gain is 225. To enhance the input accuracy and robustness, we design a set of novel schemes for *FaceInput* based on the Mel-frequency cepstral coefficient (MFCC) concept and a hidden Markov model (HMM) to process the vibration signals, and an online calibration and adaptation scheme to recover the error due to temporal instability. Extensive experiments have been conducted on 30 human subjects during the period of one month. The results demonstrate that FaceInput can be successful to sense the tiny facial vibrations and robust to fight against various confounding factors. The average recognition accuracy is 98.2%. Furthermore, by enabling the runtime calibration and adaptation scheme that updates and enlarges the training data set, the accuracy can reach 100%.**

I. Introduction

Wearable devices have become prevailing in recent days. It is forecast that the shipments of wearable devices worldwide would reach 453.19 million units by 2022 [24]. Wearable devices such as smartwatches and smart wristbands are widely used. More and more applications such as short message service(SMS) and mobile payment are adopted on the smart wristbands instead of mobile phones. Among these applications, the realization of an available typing/text entry system is essential [1]. For better user experience, the smart wristbands have become tinier and lighter, while the touch screen is getting smaller. As a result, it is difficult for users to type the correct keys.

This grand challenge has prompted considerable research efforts in the context of the mobile text entry. Several novel text-entry methods, such as FingerIO [2] and LLAP [3], have realized millimeter-scale position accuracy for fingertip tracking. As a result, users can write letters on ubiquitous surfaces instead of a tiny touch screen. However, writing letters requires operation from the non-wearing hand, which may not be available when the non-wearing hand operates other tasks. FingerT9 [4] leverages the finger segments of a user to realize the action of thumb-to-finger, which enables users to type with the same-side-hand(SSH) on the smartwatches. For the same reason, it is not available for users whose hands are fully occupied with other tasks (e.g., operating a machine or carrying objects). Furthermore, those who suffer from diseases (e.g., quadriplegic or Parkinson) have difficulty in operating their hands. Although speech recognition [9] is an alternative input method, it is sensitive to noise levels, prone to replay attacks and easy to be impersonated.

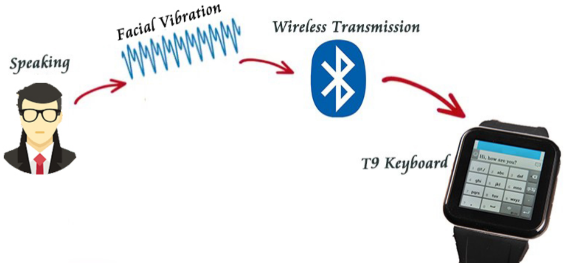


Fig. 1. A sample example of *FaceInput*.

Suffering from these pain points, we would like to seek a method to tackle the problems inherently. Especially, we have observed that the voice speech of a user can cause facial vibrations. Inherently, the facial vibration signals propagate through the face in a closed channel, which is hard to be impersonated and robust against replay attacks. The facial vibrations can resist acoustic noise and ambient dynamics. Furthermore, the facial vibration produced by voice speech is a signal source caused by a hand-free manner. One more important observation is that the facial vibrations produced by the same word exhibit highly consistent vibration fading patterns, due to vibration reflections cancelling or strengthening each other. Such patterns depend on the vibration frequency or wavelength, which can be conveniently used as vibration signatures.

Therefore, we propose *FaceInput*, as shown in Fig. 1, a lightweight wearable input system which enables users to input through facial vibration. This is implemented by relating a T9 keyboard layout to different numbers (e.g., 0,1,2,...,9). We leverage a vibration sensor on a piece of glasses to collect the facial vibration signals by speaking different numbers. Using FaceInput system, all kinds of wearable devices (e.g., smart watch) are able to receive the human text input contents through wireless transmission techniques such as Bluetooth.

Motivated by this, we have implemented a text input system,

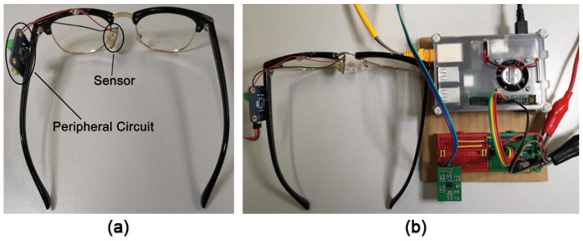


Fig. 2. (a) Vibration sensor attached to a pair of eyeglasses. (b) A sample prototype of FaceInput.)

as shown in Fig. 2(b). To realize such a reliable and usable system, we encounter the following several key challenges. First, to collect facial vibration signals produced through speech, we have to design a sensitive hardware system since the facial vibration signals on the face are too difficult to detect. Second, we have observed that users may speak with different length of the same number, which is difficult to train the constructed classification model. Thirdly, as an available system, *FaceInput* should not only work with high accuracy in an ideal environment but also in many practical situations. For example, *FaceInput* is a training system to learn the location of glasses, but user glasses could be displaced while being worn in daily life. Fourthly, *FaceInput* should work well too when users walk or shake their heads, which may produce vibration noise to the system.

To cope with these challenges, we design a double-stage amplifier whose maximum gain is 225. With the amplification by this amplifier, we can successfully collect the tiny facial vibration signals. We then studied the facial vibration propagation mechanisms on different facial vibration patterns produced by speaking operation of different numbers. We find that facial vibration propagation is a dynamic process related to the facial vibration frequency, which can be described by the Mel-frequency cepstral coefficient (MFCC) [27]. Therefore, MFCCs are able to distinguish different types of facial vibrations and can be reliably used as a vibration signature. To remove the noise signal caused by human mobility, we leverage a filter to eliminate the noise. We then use an online dual-threshold endpoint detection algorithm to detect facial vibration signals. Also, we find that the Hidden Markov Model(HMM) [28] is a suitable technique to classify vibration signals, which can address different length of voice speech. Last but not the least, to enhance the robustness in practical situations such as positional variation of glasses or different strength of voice speech, we design a runtime calibration and adaptation system and offer a special scheme to update and enlarge the training set.

We use a small form factor piezoelectric ceramic to implement *FaceInput* as a prototype on a Raspberry Pi in a real time manner. A demonstration video is shown in the link[1]. Our baseline experiment indicates that vibration classification accuracy is 98.2% on average for 30 experimenters with an initial training sample size of 10 for each number. Furthermore, we have conducted a series user studies in terms of several realistic cases such as positional variation of glasses and different voice strength. The result shows that *FaceInput* is able to recover the degradation with the runtime calibration and adaptation scheme.

To summarize, the contributions of this work consist of the following aspects.

- To the best of our knowledge, *FaceInput* is the first attempt in the literature to achieve text entry for wearable devices via the facial vibration signals which are collected by a single small size vibration sensor. It detects the facial vibrations in a closed channel to recognize the text, which is more secure and robust compared to the speech recognition schemes in an open channel.
- We implement the prototype of *FaceInput*, then leverage MFCC and HMM to recognize the facial vibrations, and adopt a runtime calibration and adaptation scheme to achieve a robust recognition system.
- We extensively evaluate the performance of *FaceInput* under different scenarios. The results show that *FaceInput* achieves an average classification accuracy of 98.2% and is robust in the different practical situations.

The remainder of this paper is structured as follows. In Section II, we first provide the background information and the related work in the context of this work. Then, Section III indicates the vibration model for facial expression. Section IV presents the system workflow and hardware prototype of *FaceInput* while showing the design goals and challenges. Section V describes the detection of the facial vibration signals. Section VI explains the recognition technique for facial vibrations. Section VII shows the detailed implementation of the runtime calibration and adaptation scheme, followed by a comprehensive experimental evaluation of our system. Finally, Section IX concludes this paper.

II. Related Work

Text input for wearable devices:

1) Opposite-side interaction on wearable devices: Nowadays, the most existing smart wristband interaction techniques leverage the input from the non-wearing hand, namely the Opposite-Side Interaction. Traditionally, text entry techniques for smart wristbands leverage QWERTY-like soft keyboards [6]. Several novel text-entry methods, such as FingerIO [2] and LLAP [3], have realized millimeter-scale position accuracy for fingertip tracking. As a result, users can write letters on ubiquitous surfaces instead of a tiny touch screen. Moreover, ViType [8] enables user to input typing on the back of the hand. However, typing on screen or the back of the hand and writing letters require operation from the non-wearing hand, which may not be available when the non-wearing hand is operating other tasks. On the contrary, *FaceInput* leverages the facial vibration to input, which can be available when the non-wearing hand is operating other tasks.

2) One-handed interaction on wearable devices: Most current smart wristband interaction methods require the input

[1] https://youtu.be/9IGplVsWWZs

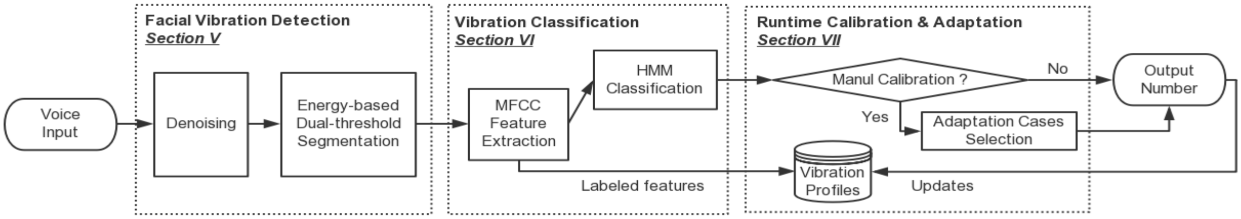


Fig. 3. The architecture of *FaceInput*.

from the non-wearing hand, but there is more and more research interest in the same-side-hand(SSH) interaction, which leverages the abilities of wrist-worn devices using the devices-worn arm/wrist. For example, Float [7] enables users to achieve one-handed and touch-free target selection on smartwatches via combining the photoplethysmogram (PPG) signal with accelerometer and gyroscope in the smartwatches. FingerT9 [4] leverages users' finger segments to realize the action of thumb-to-finger, which enables users to type with the same-side-hand on the smartwatches. However, these works are not available for users whose hands are fully occupied with other tasks(e.g., operating a machine or carrying objects). So *FaceInput* proposes a novel technique which enables users to input via the facial vibration signals.

3) Hands-free interaction on wearable devices: Speech recognition [9] is a pervasive method for text input on wearable devices. However, voice as an input scheme is inherently insecure, as it is easy to replay attacks, sensitive to noise and prone to be impersonated in an open channel. Nowadays, studies have indicated that intruders can inject their voice commands stealthily and remotely with mangled voice [10], wireless signals [11] or through public radio stations [12] without causing users' attention. Different from speech recognition, *FaceInput* using the facial vibration signals for input works in a closed channel, which is difficult to inject the vibration signals stealthily and remotely.

Sensing technologies for facial activity: Nowadays, there are various techniques proposed to sense facial expressions. In the following, we offer a brief overview of these techniques.

1) Optical sensing: The most existing technique is using a vision-based camera to track users' facial expressions [13]. However, a vision-based camera tracking technology is prone to be affected by lighting conditions. In addition, camera-based detection systems are usually bulky or stationary. Furthermore, they are prone to be invaded for personal information. *FaceInput* uses facial vibration signals for text input, which can not be affected by ambient environment and does not let out personal information.

2) Electromyography(EMG): The most essential action is a binary on/off-switch, that can be conducted by sensing an emerging action potential, such as produced by contracting muscles. It has been indicated by San Agustin using an EMG headband that senses a frowning or a tightening of users' jaw [14]. However, these works have to attach extra devices to the users' skin, which is invasive, obtrusive and unacceptable. In contrast, *FaceInput* senses the facial vibration signals via a piezoelectric ceramic vibration sensor which can be attached to the glasses of a user.

3) Electroencephalography(EEG): With EEG technology, we can measure neuro-activity signals on the cortical surface or within the brain, namely Brain Computer Interfaces (BCIs). Matthies et al. [15] used eye winking, ear wiggling, and head gestures to operate a handheld via an Emotiv's mobile EEG headset. However, an EEG headset is bulky and hardly acceptable in realistic situations. Furthermore, EEG technology also requires users to attach EEG sensors to their skin, which is obtrusive.

4) Electrooculography(EOG): With EOG Glasses, we can basically detect eye-movements, which can control smart environments [16]. Ishimaru et al. [17] utilized EOG technology to roughly recognize chewing, talking, eating, and reading with an average accuracy of 70%. As described above, EOG sensors only can roughly identify the facial activities, which can not sense the tiny facial vibration signals created by speaking words.

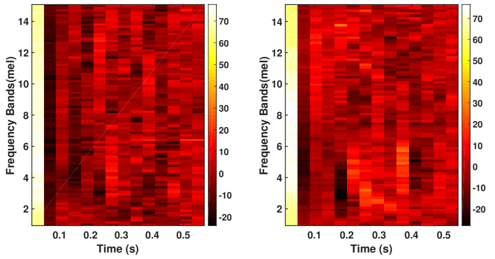
5) Capacitive sensing(CS): Rantanen et al. [18] proposed a CS-based glass which could detect eyebrows' frowning and lifting to perform click-events. Furthermore, Rantanen et al. [19] presented a face-hugging device consisting of 12 electrodes. They observed that the activation of four different muscle groups could be sensed via a proximity sensing. However, it requires users to wear a face-hugger that covers the entire face of a user, which is rather obtrusive.

6) Electromagnetic sensing: Fagan et al. [20] attached 7 magnets onto the lips, teeth, and tongue which could create a significant change in the magnetic field when users conducted mouth-movements. And there were 6 Dual axis magnetic sensors mounted on the glasses, which could detect 13 phonemes and 9 words. However, the experiment setup was quite bulky and obtrusive.

7) NAM microphone(stethoscopic microphone): Chen Jou et al. [21] utilized a throat microphone to recognize whisper words. Panikos Heracleous et al. [22] presented the use of a stethoscope and silicon NAM (nonaudible murmur) microphones in automatic speech recognition. However, these works had to use NAM Microphone to detect the voice signals, which could be affected by ambient noise.

III. Vibration Model for Facial Expression

Human speech: The production of human speech is widely referred to as the source-filter model [26], which consist of two separate processes: 1) The original sound source is first



(a)The extracted MFCC features of key "zero" (b)The extracted MFCC features of key "one"

Fig. 4. Example of the extracted MFCC features.

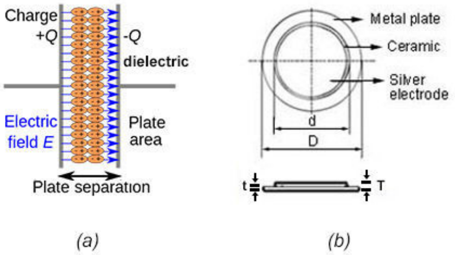


Fig. 5. A sample piezoelectric ceramic.

generated through the vibration of vocal folds. 2) Then the sound source will be filtered and modulated when traveling through the vocal tract. The output sound will be shaped into different spectrum when the structure of vocal tract varies (e.g., the tongue movement).

Multipath channel profile-based signature: Fig. 7(a) plots the facial vibration signals captured by the piezoelectric ceramics when a user speaks a word (i.e., "taps"). The magnitude and duration of different phases(e.g., s1, s2, s3) show obvious dissimilarity. Further, we can see from Fig. 4 that the corresponding MFCC profiles of voice "zero" and "one" exhibit different values across the frequency bands and time. Hence, the MFCC features can be utilized to characterize the user's facial vibrations.

IV. *FaceInput* System Design

A. *Design goals and challenges:*

The main objective of this work is to propose a hand-free and secure text input system through the facial vibration signals. Thus *FaceInput* is designed to meet the following goals.

1) Availability: FaceInput should function properly in most of the daily user scenarios while inputting text. It should be resilient to the surrounding acoustic noise and the human-motion noise while walking or shaking the head.

2) Robustness: The users may pronounce the same word with different volume, tone, and duration, which results in variation of facial vibration signals with respect to amplitude, frequency and length. Therefore, *FaceInput* should be robust enough to give the correct output when this regular variation of input signals occurs.

3) Efficiency: FaceInput should be efficient with low time and computation overhead. Specifically, we have to make sure

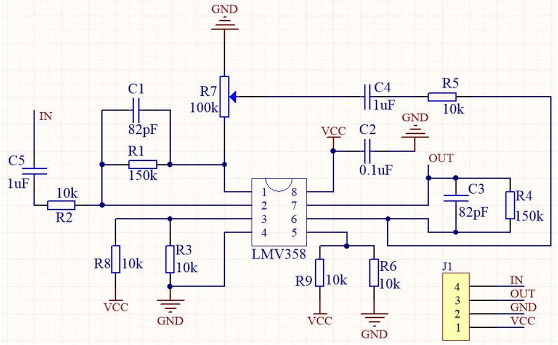


Fig. 6. The design of the AC amplifier circuit.

that users do not require to rebuild the vibration profile each time when they launch the system. Such time overhead is catastrophic to the user when the usage duration is short. Further, the computation overhead should also be as low as possible for a low latency text-input system.

B. *System workflow:*

In Fig. 3, *FaceInput* records the vibration signals using a small form factor piezoelectric ceramic which is small enough to be embedded into the glasses. After receiving the vibration signals in the time domain, the system applies a series of signal processing operations to obtain a useful segment of facial vibration signals (Section V). Then, the segmented signals are transformed to matrices with unique features (i.e., Mel-Frequency cepstral coefficient), which corresponds to the vibration characteristics of the input number. In the profiling stage, these features are labeled and saved to build the vibration profiles of each number (i.e., 0-9), which is used to construct a Hidden Markov Model. In the recognition stage, the subsequent facial vibration samples are taken as inputs to the pre-trained Hidden Markov Model. The model compares the new input with the vibration profiles and provides the recognition result based on statistical calculation. Finally, *FaceInput* has a runtime adaptation mechanism to calibrate the occasional recognition errors.

C. *Hardware prototype*

In Fig. 2(a), a piezoelectric ceramic vibration sensor is attached to a pair of eyeglasses, which is used to sense the facial vibration signals. The piezoelectric ceramic is a vibration sensor that uses the piezoelectric effect, to measure the facial vibrations, by converting it to an electrical charge. Fig. 5(a) shows a standard dielectric in a capacitor. In a piezoelectric device, mechanical stress, instead of an externally applied voltage, causes the separation of charge in the individual atoms of the material. Thus, the vibrations caused by talking can be converted to an electrical charge. Fig. 5(b) shows a sample piezoelectric ceramic whose external diameter D is 20mm and thickness T is 0.35mm. The small size allows us to embed it to wearable devices like smart glasses. We place the sensor at the position as shown in Fig. 2(a) as this area has consistent contact with the face of a user.

Besides, if we desire to extract facial vibrations in the presence of noise caused by body movements while walking or

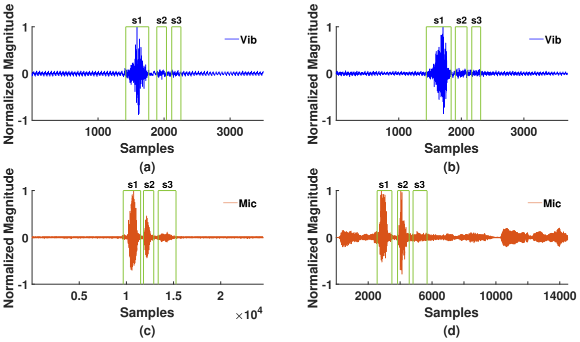


Fig. 7. Sample raw input signals of a word(i.e., “taps”).

shaking the head, the amplifier should be kept small enough to avoid the noise of body movement. On the other hand, we are limited by the resolution of the ADC unit, especially that with low-cost. If the amplification is too small, it becomes hard to correctly detect facial vibration due to a combination of low signal amplitude and low ADC resolution (i.e., quantization error becomes dominant). We have configured the amplifier circuit carefully to ensure *FaceInput* is sensitive enough to detect even the tiny input command from the user. To this end, we require to make sure that the facial vibration signals caused by human speech stay within the range of the amplifier circuit output after amplification, i.e., 0-5V in our case.

Fig. 6 shows the design of the AC amplifier circuit, which is a double-stage amplifier whose maximum gain is 225. To filter the system noise by a hardware circuit, both the stages of the amplifier circuit have an RC bandpass filter in the 15.9Hz to 12.9kHz range. Also, the maximum gain of both the stages is 15. When the signal is amplified by the first stage, the amplification of the signal can be adjustable by turning the adjustable resistor R_7 shown in Fig. 6. Therefore, the maximum gain of this amplifier circuit is 225. The amplified signal is in the range from 0 to 5V, which is quantized to 1024 levels (10 bits) using an MCP3008 A/D converter [25]. The ADC output signal is thus transmitted to a laptop for further processing of signal and extraction of facial vibration.

V. Detection of Facial Vibration Signal

A. Denoising

Unlike a microphone-based acoustic system using the vibration signals, *FaceInput* is capable of resisting environmental acoustic noise. Fig. 7(a) and Fig. 7(c) show the facial vibration signals and voice signals of a word (i.e., “taps”) with 50dB ambient noise. In addition, Fig. 7(b) and Fig. 7(d) show the facial vibration signals and voice signals of a word (i.e., “taps”) with 90dB ambient noise. Fig. 7 shows that although piezoelectric ceramic sensors are not sensitive to detect the speech as good as microphones, they are naturally against the surrounding noise. We will introduce that *FaceInput* is able to recognize speech via the facial vibrations in section VI. Through the recognition of vibrations in a closed channel, *FaceInput* is more secure, which is against to impersonate and replay attacks. However, piezoelectric ceramic sensors are sensitive to human mobility, such as walking or shaking the head.

We adopt the Fast Fourier Transform (FFT) technique to analyze the frequency between human mobility and facial vibrations. The result shows that the frequency of human mobility is always under 10 Hz while the frequency of facial vibrations is in the 10 to 1000 Hz range. We observe that there is a clear separation between facial vibrations and body movements in the frequency domain. Therefore, we use a Butterworth bandpass filter to denoise body movement and high-frequency noise in the 10 to 1000 Hz range.

B. Segmentation

After denoising the vibration signals, *FaceInput* adopts an energy-based dual-threshold scheme [5] to detect the start and end points of the useful facial vibration signals. Specifically, it calculates the energy levels by accumulating the square of the amplitudes of received vibration signals in a sliding window in the time domain as follows.

$$Energy(t) = \sum_{i=t}^{t+L} s^2(i) \tag{1}$$

where L is the length of the sliding time window and $s(i)$ is the amplitude of the received vibration signals. As for the low threshold TL and high threshold TH, we calculate the mean of noise energy and standard deviation of signal energy as μ and σ respectively, while setting $TL = \mu + \sigma$, $TH = \mu + 3\sigma$.

VI. Vibration Classification

A. Feature extraction

Due to represent the dynamic features of the signals with both linear and nonlinear properties, the Mel-frequency cepstral coefficient (MFCC) is widely applied to represent the short-term power spectrum of acoustic or vibration signals [27]. While the MFCCs are preferred choice for researchers to tackle the voice recognition task, we observe that they can also characterize the facial vibration signals caused by speaking, as the vibration signals generate different vibration energy at different frequencies and propagate over different distances on the face. Specifically, we calculate the MFCCs of the received vibration signals in each sliding window. The number of filterbank channels is set to 28, and 14-th order cepstral coefficients are computed in each 20 ms Hamming window, shifting 6 ms each time.

B. Classification

The hidden Markov model is a statistical tool to capture the time series structure of the facial vibration signals, which is regarded as a mathematical double stochastic process [29]. One is to use the Markov chain with a finite state number to simulate the implicit stochastic process of the statistical properties of the facial vibration signals. The other is a random process of observation sequences associated with each state of the Markov chain. Note that the former one is expressed by the latter, but the specific parameters of the former are unmeasurable. We observe that the human facial vibration process is actually a double random process, as the facial vibration signals are observable time-varying sequences, which are produced by the brain based on grammatical knowledge and verbal requirements.

Furthermore, the grammatical knowledge and verbal requirements can be regarded as unobservable states. It can be seen that the HMM reasonably imitates this process and describes the overall non-stationary and local stationarity of the facial vibration signals. (e.g., variational pronunciation length of a word.)

A Hidden Markov Model $\boldsymbol{\lambda}$ is represented as

$$\lambda = (A, B, \pi) \tag{2}$$

where $\boldsymbol{A}$ is the state transition matrix, $\boldsymbol{B}$ is the set of emission probability distributions, and $\boldsymbol{\pi}$ is the initial state probability vector. Given any time $\boldsymbol{n}$, an observation is associated with one of the discrete hidden states. The random variable representing the hidden state at time $\boldsymbol{n}$ is denoted by $\boldsymbol{q_n}$, while the random variable representing the observation at time $\boldsymbol{n}$ is denoted by $\boldsymbol{p_n}$. In the state transition matrix $\boldsymbol{A}$, each element $\boldsymbol{a_{ij}}$, which indicates the transition probability from state $\boldsymbol{i}$ to state $\boldsymbol{j}$ and it is defined as

$$a_{ij} = P(q_n = j | q_{n-1} = i) \tag{3}$$

Given the hidden state q_n, the probability of an observation vector, i.e. the emission probability is

$$b_n^i = P(v_n = O_n | q_n = i) \tag{4}$$

Then the set of emission probabilities $\boldsymbol{B}$ is denoted as

$$B = \{b_n^i\}, \qquad i = 1, 2, ...S \tag{5}$$

where S is the number of discrete hidden states, and it is set to 3 in our case.

Note that FaceInput will be applied to wearable devices with limited computing resources, which requires our training process to be efficient enough and less costly to train. Therefore, to reduce the training time of the system and the cost of computing resources, FaceInput uses the Baum-Welch algorithm [31] to train the parameters instantaneously instead of multiple iterations. Besides, FaceInput utilizes Viterbi algorithm [32] to evaluate the signal samples, which need to perform logarithmic operations on the starting probability and the transition probability. In order to prevent the underflow caused by the logarithm of 0, FaceInput first finds the elements with probability less than or equal to 0, and then directly assigns them a very small negative number instead of the one taken from the logarithm directly.

VII. Online Calibration and Adaptation

FaceInput works based on the assumption that the facial vibration signals remain stable throughout its usage life-cycle. In the practical scenarios, the signal patterns may be disturbed by, e.g., users' repositioning of the glasses. Over the time, users speaking postures may also vary due to physiological status, that renders MFCC features in the initial training set outdated. As a result, *FaceInput* leverages a run-time calibration and adaptation scheme to address these problems.

A. Runtime calibration

FaceInput conducts the run-time calibration by user's correction and its own classification hints. As for calibration, for each sample, besides the output from the classification algorithm, it also recommends the top 2 candidate keys, i.e., those with the largest possibility which are shown on the touch screen of smart watch. A user can click a candidate key if it is the actually intended key when the classification algorithm gives an erroneous output on the touch screen.

In the rare case, the user has to reenter the key while leveraging the built-in on-screen keyboard when the intended key is not in the candidate list. To be reliably recognized for calibration purpose, the "Delete" key is placed on the screen instead of input through voice.

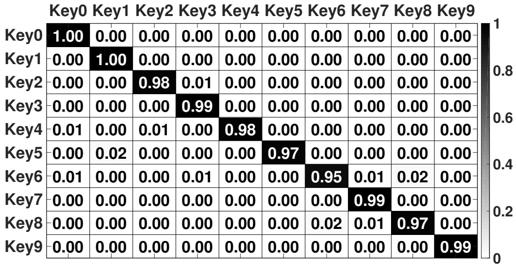
B. Adapting and optimizing training set

In terms of practical usage, *FaceInput* updates the training data set progressively in four cases: (1) *FaceInput* deems the classification output as correct if the user does not click the candidate key, (2) the user clicks any candidate key, which implies that a classification error occurs and the intended key is contained in the candidate list, (3) the user tapping a key by using the built-in keyboard implies that a classification error occurs and the intended key is not contained in the candidate list, (4) the user clicks the "Delete" button on the screen, which is not necessarily a hint for classification error since it may be the user's own input error.

Therefore, for adapting and optimizing training set, we have constructed an adjustable scheme to update the training data set in a different case. For case 1, the input instance will be added only once into the temporary queue that corresponds to the correct classification output. For case 2, the input instance is added n_i times into the temporary queue corresponding to the selected candidate key. Note that n_i is defined as continuous error times of key i and varies from 1 to 3. For example, if the classification algorithm gives the erroneous output of key i for 2 times, the value of n_i is set to 2. If the erroneous output of key i occurs more than 3 times continuously, the value of n_i will be 3. And once the classification algorithm gives the correct output for key i, n_i will be reset to 1. For case 3, the input instance is added 3 times into the temporary queue corresponding to the selected candidate key. For case 4, the input instance is discarded instead of being added to the temporary queue. We define the number of instance in the temporary queue of key i as Q_i. Thus, we can obtain the total number of instances in all temporary queues

$$N = \sum_{i=0}^{9} Q_i \tag{6}$$

Once N is larger than 10, the instances in all temporary queues are added into the training data set, and the HMM model is trained again. Meanwhile, all temporary queues are cleared. Note that the oldest instances will leave the training data set, while the training set size of the corresponding key reaches the maximum of 35.



	Key0	Key1	Key2	Key3	Key4	Key5	Key6	Key7	Key8	Key9
Key0	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Key1	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Key2	0.00	0.00	0.98	0.01	0.00	0.00	0.00	0.00	0.00	0.00
Key3	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.00	0.00
Key4	0.01	0.00	0.01	0.00	0.98	0.00	0.00	0.00	0.00	0.00
Key5	0.00	0.02	0.00	0.00	0.00	0.97	0.00	0.00	0.00	0.00
Key6	0.01	0.00	0.00	0.01	0.00	0.00	0.95	0.01	0.02	0.00
Key7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00
Key8	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.01	0.97	0.00
Key9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.99

Fig. 8. Confusion matrix of 10 keys.

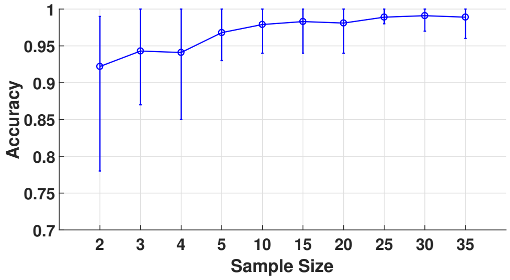


Fig. 9. Impact of initial training set size for FaceInput.

VIII. Implementation & Evaluation

Implementation: In our prototype, we leverage a piezoelectric ceramic sensor attached into a piece of glasses to collect the facial vibration signals with the 2 kHz sampling rate. The vibration signals are amplified by an amplifier which is connected to a Raspberry Pi controller using MCP3008 [25], an Analog to Digital Converter (ADC). The whole data acquisition process is implemented via BCM2835 Library [23] with C. Then, the collected data are transmitted to a conventional laptop by a network cable, which is implemented via the TCP protocol.

As for the number classification, the signal denoising, facial vibration signals detection and HMM algorithm are implemented in Matlab. Note that the number status of the HMM is 3 and the number of Gaussian probability-density function of each status is 2.

Experimental setup: We recruited 30 participants(18 of them are male) whose age is in the [18-23] range and body mass indexes (BMIs) range from 16.25 (thin) to 30.36 (obese) as they represent the crowds that are most possibly to use our system.

The evaluation experiments are conducted in a conventional office environment. Before the experiments, participants are given 2 minutes to become familiar with our system. In all experiments, we leverage the following default setting unless otherwise specified. The participants are instructed to speak 20 times on each key in an orderly fashion (200 samples for each person, and thus 6000 samples in total). For example, we ask the participants to speak key 1 for 20 times, then key 2 for 20 times, and so on. We then randomly select 10 samples from each key (100 samples for a person) to initialize the HMM learning model to estimate the mathematical model. We leverage the other 10 samples left to test the classification accuracy of the HMM. We repeat the process above 20 times, and thus we can obtain the average classification accuracy of the HMM.

A. Accuracy of *FaceInput*

1) Baseline detection and classification: We first evaluate the detection and classification of inputs to establish a baseline performance of *FaceInput*. We asked 30 participants to input numbers from ”zero” to ”nine” for 20 rounds by speaking in an ordinary office environment. The confusion matrix in Fig. 8 shows the classification performance of *FaceInput* with 10 training samples for each number. The diagonal in the matrix represents the number of points for which samples are correctly classified. We notice that *FaceInput* achieves an average classification accuracy of 98.2% and the variance of the accuracy is about $2.5e-04$.

2) Impact of training set size: To evaluate the case associated with enlargement of the training set size, 10 of the participants were asked to speak each number for 45 times. Fig. 9 plots the resulting variance of accuracy as the training set size increases from 2 to 35. We observe that classification accuracy monotonically increases as the training set size increases. The accuracy of *FaceInput* escalates to about 98% as the training set size grows up to 10. Moreover, while increasing the number beyond 25, the accuracy further approaches 100%. In reality, it is achievable since the runtime adaptation mechanism of *FaceInput* can enlarge the training set size and provide the optimal performance of nearly 100% classification accuracy as the users speak continuously within a short usage time.

B. Robustness of *FaceInput*

1) Positional variation of glasses: *FaceInput* presumes that the glasses are stable in place to provide vibration signals of good quality when user input different words by speaking. However, it is quite normal that worn glass of a user shifts slightly over the time. To evaluate the impact on the positional variation of glasses, we asked 10 participants to speak each number for 20 times. The participants were asked to repeat after shifting the glasses downwards $1cm$ to P_2 from the original position P_1. Therefore, in Fig. 10(a), “1-2” means that we take 10 samples of P_1 to train the HMM and then test with 10 samples of P_2. In particular, “A-A” means that the samples collected from P_1 and P_2 are both in the training and testing set (e.g., 10 samples from P_1 and 10 samples from P_2 for training, and the test set resembles the training set.). From the results shown in Fig. 10(a), we observe that the samples from the same position (e.g., 1-1 and 2-2) obtain much higher accuracy than the cases of “1-2” and “2-1”. Further, if we train the system with the samples from different positions (e.g., A-A), *FaceInput* can achieve an accuracy of 98.95%, which means that we can eliminate the slight degradation resulted by the displacement of the glasses with a runtime adaptation scheme which can update the system training set.

2) Voice strength: Even for saying the same word, the resultant vibration signals can be different because of different voice strength. To investigate how the system performance

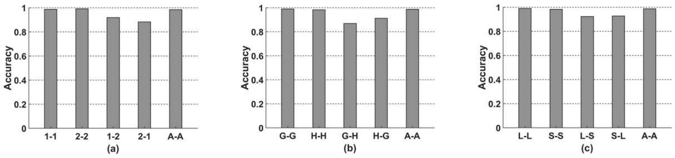


Fig. 10. (a) Accuracy of positional variation of glasses. (b) Impact of different voice strength. (c) Impact of different voice length.

is influenced by voice strength, 10 participants were asked to say words both heavily and gently. Fig. 10(b) shows the classification accuracy under different voice strength, and we note that here "H" represents a user speaks heavily while "G" represents a user speaks gently. Still, the label of X-axis is in the format of "training sample - test sample". We observe that the classification accuracy decreases to around 87% when the voice force of test sample is different from that of the training set like the cases of "G-H" and "H-G". Fortunately, our runtime adaptation scheme can record both the gentle and heavy cases to update the vibration profiles. Consequently, training set will resemble the "A-A" model achieving a high accuracy up to 99% when users apply different voice strength in daily input.

3) Voice length: As we all know, a critical challenge in voice recognition system is to recognize the input as the single correct word even if the voice duration of input is variant. Thus, in this experiment, we investigate whether *FaceInput* can tackle the length variation challenge of voice and ask 10 participants to use *FaceInput* with different voice lengths. The resultant classification accuracies are shown in Fig. 10(c), from which we can observe that *FaceInput* is robust against different voice lengths (L: long, S: short, A: all). The reason for high accuracy is that we adopt MFCC (Mel Frequency Cepstral Coefficient) to be feature parameters, and train the classifier for each word following a Hidden Markov Model (HMM). MFCC can represent the dynamic features of the signals with both linear and nonlinear properties, which is widely applied to represent the short-term power spectrum of acoustic or vibration signals [27]. In the HMM, the profiles of each key are stored as a state diagram instead of time series, which can resist the length variation of voice.

4) Mobility: There is a common scenario that the users require to reply to a message while walking or shaking the head. We are interested in how the system performs with the occurrence of noise interference caused by physical movements. Table I lists the average recognition accuracy when participants input the numbers and walk or shake their heads simultaneously in an ordinary office environment. By comparing the results with the baseline, we can see that there is almost no influence on system performance. The reason is that we remove the low-frequency noise of human mobility ($\leq$ 10Hz) through a Butterworth bandpass filter.

C. Runtime calibration and adaptation

We design the runtime calibration and adaptation scheme to maintain high recognition accuracy and make the system more robust and resilient under different practical scenarios.

TABLE I
Classification accuracy with respect to human mobility.

Items	Standing(Baseline)	Walking	Shaking the head
Accuracy	98.6%	94.9%	97.1%

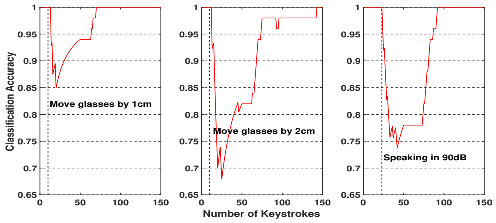


Fig. 11. The runtime calibration and adaptation scheme helps restoring high accuracy of *FaceInput*. Dotted lines denote the occurrence moment of variation.

In the following experiments, we turn the runtime calibration and adaptation scheme on in a runtime demo. We count the average recognition accuracy over the last 50 inputs in terms of the variation of glasses position and input volume size. From Fig. 11, we can draw the following conclusions.

1) Resilience to displacement: In terms of the displacement of glasses position, we move the glasses downwards 1 cm and 2 cm from the original position, and the accuracy declines to about 85% and 68% respectively. Under the assistance of runtime adaptation scheme, the accuracy recovers quickly after a few tens of inputs.

2) Resilience to volume change: When there are only low volume samples (at 75 dB) in the training set, we can get good recognition accuracy using low volume test samples. However, if the volume of the input command bursts up to 90 dB, the recognition accuracy drops to around 75%. Note that the volume level is measured by an Android application: Sound Meter Pro [30]. Similarly, the runtime adaptation scheme restores the system performance in a short time.

D. Temporal stability

To validate the temporal stability of *FaceInput*, we conduct the same experiments 5 times throughout an hour, 1 day, 2 days, 1 week and 1 month. The glasses are fixed in the same position according to the user's habit. In each time, we spoke from key "zero" to key "nine" for 100 rounds (1000 samples in total). And we recorded the average classification accuracy of the last 50 samples. With the enlargement of training samples size, the classification accuracy remained stable at around 98% each time. This indicates that FaceInput is temporally stable over the time.

E. Cost

For time overhead, *FaceInput* requires users to input 10 × 10 training samples to initialize the HMM, and all the users can finish the input within 3 minutes based on the statistic. After the user input phase, 2.2s is required to train the HMM. Furthermore, on average, the latency between the voice inputs and the outputs is 0.25s. Therefore, there is no lagging effect during the usage duration since the latency is below the human response time. For hardware cost, *FaceInput* deploys only one piezoelectric ceramic that costs 0.15 dollar, which is inexpensive for manufacturers to embed *FaceInput* on a glass.

IX. Conclusion

In this paper, we propose a novel hand-free and secure text-input system for the smartwatch by mapping the facial vibrations generated from human speech. The facial vibration is detected by a small piezoelectric sensor embedded on the glasses and then the input number is estimated by the Hidden Markov Model. We conduct extensive experiments with the voice commands from a set of participants. The results indicate that *FaceInput* achieves high recognition accuracy for ten kinds of commands with the accuracy of 98.2%. It is resilient to environmental acoustic noise and will not be disturbed by the voice commands of other users. Furthermore, it shows strong robustness under several daily text-input scenarios.

Acknowledgment

This research was supported in part by the China NSFC Grant (61872248, U1736207), Guangdong NSF 2017A030312008, Shenzhen Science and Technology Foundation (No. JCYJ20170302140946299, JCYJ20170412110753954), Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China(Grant No.161064), GDUPS (2015). This work is partially supported by Tianjin Key Laboratory of Advanced Networking (TANK), School of Computer Science and Technology, Tianjin University, Tianjin China, 300350. Kaishun Wu is the corresponding author.

References

[1] K. Zhu, X. Ma, H. Chen, and M. Liang (2017). Tripartite Effects: Exploring Users Mental Model of Mobile Gestures under the Influence of Operation, Handheld Posture, and Interaction Space. The International Journal of HumanComputer Interaction, Vol. 33, No. 6 (pp. 443-459).

[2] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota (2016). Fingerio: Using active sonar for fine-grained finger tracking. In Proc. ACM CHI (pp. 1515-1525).

[3] W. Wang, A. X. Liu, and K. Sun (2016, October). Device-free gesture tracking using acoustic signals. In Proc. ACM MobiCom (pp. 82-94).

[4] P. C. Wong, K. Zhu, and H. b.Fu (2018). FingerT9: Leveraging thumb-to-finger interaction for same-side-hand text entry on smartwatches. In Proc. ACM CHI (pp. 178).

[5] W. Chen, M. Guan, L. Wang, R. Ruby, K. Wu.(2017, July). FLoc: Device free passive indoor localization in complex environments. In Proc. IEEE ICC (pp. 1-6).

[6] L. A. Leiva, A. Sahami, A. Catala, N. Henze, and A. Schmidt (2015). Text entry on tiny qwerty soft keyboards. In Proc. ACM CHI (pp. 669-678).

[7] K. Sun, Y. Wang, C. Yu, Y. Yan, H. Wen, and Y. Shi (2017). Float: One-Handed and Touch-Free Target Selection on Smartwatches. In Proc. ACM CHI (pp. 692-704).

[8] W. Chen, M. Guan, Y. Huang, L. Wang, R. Ruby, W. Hu, and K. Wu (2018, June). ViType: A Cost Efficient On-Body Typing System through Vibration. In Proc. IEEE SECON (pp. 1-9).

[10] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou (2016). Hidden Voice Commands. In USENIX Security Symposium (pp. 513530).

[9] V. Lakshmipathy, C. Schmandt, and N. Marmasse (2003, November). TalkBack: a conversational answering machine. In Proc. ACM UIST (pp. 41-50).

[11] C. Kasmi and J. L. Esteves (2015). IEMI Threats for Information Security: Remote Command Injection on Modern Smartphones. In IEEE Transactions on Electromagnetic Compatibility, Vol. 57, No. 6 (pp. 17521755).

[12] R. Martin (2016). Listen Up: Your AI Assistant Goes Crazy For NPR Too.
http://www.npr.org/2016/03/06/469383361/listen-up-your-ai-assistant-goescrazy-for-npr-too.

[13] B. Fasel, and J. Luettin (2003). Automatic facial expression analysis: a survey. In Pattern recognition, Vol. 36, No. 1 (pp. 259-275).

[14] J. S. Agustin, J. P. Hansen, D. W. Hansen, and H. Skovsgaard (2009). Low-cost gaze pointing and EMG clicking. In Proc. ACM CHI (pp. 3247-3252).

[15] D. J. C. Matthies, J. N. Antons, F. Heidmann, R. Wettach, and R. Schleicher (2012). NeuroPad: use cases for a mobile physiological interface. In Proc. ACM NordiCHI (pp. 795-796).

[16] A. Bulling, D. Roggen, and G. Trster (2009). Wearable EOG goggles: eye-based interaction in everyday environments, In Proc. ACM CHI (pp. 3259-3264).

[17] S. Ishimaru, K. Kunze, Y. Uema, K. Kise, M. Inami, and K. Tanaka (2014). Smarter Eyewear: using commercial EOG glasses for activity recognition. In Proc. ACM Ubicomp (pp. 239-242).

[18] V. Rantanen, P. H. Niemenlehto, J. Verho, and J. Lekkala (2010). Capacitive facial movement detection for humancomputer interaction to click by frowning and lifting eyebrows. In Springer Medical and biological engineering and computing, Vol. 48, No. 1 (pp. 39-47).

[19] V. Rantanen, H. Venesvirta, O. Spakov, J. Verho, A. Vetek, V. Surakka, and J. Lekkala (2013). Capacitive measurement of facial activity intensity. In IEEE Sensors Journal, Vol. 13, No. 11 (pp. 4329-4338).

[20] M. J. Fagan, S. R. Ell, J. M. Gilbert, E. Sarrazin, and P. M. Chapman (2008). Development of a (silent) speech recognition system for patients following laryngectomy. In Medical engineering and physics, Vol. 30, No. 4 (pp. 419-425).

[21] S. C. Jou, T. Schultz, and A. Waibel (2004). Adaptation for soft whisper recognition using a throat microphone. In Proc. INTERSPEECH.

[22] P. Heracleous, T. Kaino, H. Saruwatari, and K. Shikano (2006). Unvoiced speech recognition using tissue-conductive acoustic sensor. EURASIP Journal on Advances in Signal Processing, Vol. 2007, No. 1 (pp. 094068).

[23] C library for Broadcom BCM 2835 as used in Raspberry Pi
https://www.airspayce.com/mikem/bcm2835/

[24] Gartner Says Worldwide Wearable Device Sales to Grow 26 Percent in 2019
https://www.gartner.com/en/newsroom/press-releases/2018-11-29-gartner-says-worldwide-wearable-device-sales-to-grow-

[25] Raspberry Pi Analog to Digital Converters
https://learn.adafruit.com/raspberry-pi-analog-to-digital-converters/mcp3008

[26] L. Haskins (1999). The Acoustic Theory of Speech Production: the source-filter model.
http://www.haskins.yale.edu/featured/heads/mmsp/acoustic.html

[27] F. Zheng, G. Zhang, and Z. Song (2001). Comparison of different implementations of MFCC. Journal of Computer science and Technology, Vol. 16, No. 6 (pp.582-589).

[28] H. Ocak, and K. A. Loparo (2001). A new bearing fault detection and diagnosis scheme based on hidden Markov modeling of vibration signals. In Proc. IEEE ICASSP, Vol. 5 (pp. 3141-3144).

[29] K. Grobel, and M. Assan (1997, October). Isolated sign language recognition using hidden Markov models. In Proc. IEEE Computational Cybernetics and Simulation, Vol. 1 (pp. 162-167).

[30] Sound Meter PRO
https://play.google.com/store/apps/details?id=com.soundmeter.app

[31] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun (1992, March). A practical part-of-speech tagger. In Proc. Applied natural language processing (pp. 133-140).

[32] G. D. Forney (1973). The viterbi algorithm. In Proc. IEEE, Vol. 61, No. 3 (pp. 268-278).