2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)

VPad: Virtual Writing Tablet for Laptops Leveraging Acoustic Signals

Li Lu*, Jian Liu†, Jiadi Yu*‡, Yingying Chen†, Yanmin Zhu*, Xiangyu Xu*, Minglu Li*
*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, P.R.China
Email: {luli_jtu, jiadiyu, yzhu, chillex, mlli}@sjtu.edu.cn
†Department of Electrical and Computer Engineering, Rutgers University, NJ, USA
Email: jianliu@winlab.rutgers.edu, yingche@scarletmail.rutgers.edu
‡Corresponding author

***Abstract*—Human-computer interaction based on touch screens plays an increasing role in our daily lives. Besides smartphones and tablets, laptops are the most popular mobile devices used in both work and leisure. To satisfy requirements of many emerging applications, it becomes desirable to equip both writing and drawing functions directly on laptop screens. In this paper, we design a virtual writing tablet system, *VPad*, for traditional laptops without touch screens. VPad leverages two speakers and one microphone, which are available in most commodity laptops, for trajectory tracking without additional hardware. It employs acoustic signals to accurately track hand movements and recognize characters user writes in the air. Specifically, VPad emits inaudible acoustic signals from two speakers in a laptop. Then VPad applies *Sliding-window Overlap Fourier Transformation* technique to find Doppler frequency shift with higher resolution and accuracy in real time. Furthermore, we analyze frequency shifts and energy features of acoustic signals received by the microphone to track the trajectory of hand movements. Finally, we employ a stroke direction sequence model based on possibility estimation to recognize characters users write in the air. Our experimental results show that VPad achieves the average trajectory tracking error of only $1.55cm$ and the character recognition accuracy of above 90% merely through two speakers and one microphone on a laptop.**

***Index Terms*—acoustic signals, laptops, virtual writing, trajectory tracking**

I. Introduction

Touch screens become a nonseparable part in smart devices, and there is a growing trend with more applications requiring interactions with devices through touch screens. Nowadays 97% smart devices are equipped with touch screen [1]. This trend has been spread into traditional laptops as laptops are the most popular mobile devices in both work and leisure besides smartphones and tablets. However, traditional laptops are not equipped with touch screens. Although small touchpads on laptops provide scrolling and swiping functionality, they lack writing and drawing capabilities, which become increasingly important in many new applications, such as WRITEit. Thus, it is essential to enable laptops with touch capability. Moreover, although some users intend to use keyboard for input, there are several situations, in which it is hard for users to input via keyboard. For example, when users are in some transportation vehicles, they cannot expediently input with keyboard due to the vibration of vehicles. For disabled people, such as a person without several fingers, traditional input approach like keyboard is difficult to interact with computers [2], which brings the necessity to enable traditional laptops with touch capability so that disabled people can conveniently interact with computers through hand gestures.

Recent products, such as Kinect [3], have demonstrated that the hand gesture is a novel way to interact with computers. However, these vision-based approaches are sensitive to the ambient lights and suffer significant performance degradation under dark environments. Recently, SoundWave [4] utilizes Doppler effect of acoustic signals to recognize hand gestures without trajectory tracking. AAmouse [5] and CAT [6] realize the accurate trajectory tracking based on acoustic signals, but both of them need additional audio devices (such as a smartphone) as an acoustic-signal emitter. LLAP [7] and FingerIO [8] propose gesture tracking schemes using acoustic signals for wearable devices, and Strata [9] develops a fine-grained acoustic-based tracker for smartphones, all of which cannot be adopted in commodity laptops because there are different audio components in laptops with that in wearable devices and smartphones. Our goal is to accurately track trajectory in real time with the audio components including a microphone and two speakers, which are available on most off-the-shelf laptops.

In this work, we take one step forward to develop a device-free virtual writing tablet (*VPad*) leveraging existing audio devices on traditional laptops without any additional hardware. By leveraging acoustic signals emitted from the laptop, VPad seeks to achieve the fine-grained trajectory tracking and accurate character recognition. To enable the virtual writing capability in the air leveraging acoustic signals, a number of challenges arise in practice. Firstly, the acoustic signal sampling rate is limited by laptops' hardware. The acoustic signals can be easily affected by ambient noise, resulting in measurement errors and instability of recorded signals. Secondly, the audio devices of laptops are constrained to two speakers and one microphone, providing limited information to perform accurate hand movement tracking. Finally, the system needs to deal with different writing habits and provide accurate character recognition based on hand movement trajectory.

To achieve two critical factors in hand movement track-

978-1-5386-7308-9/18/$31.00 ©2018 IEEE
DOI 10.1109/ICPADS.2018.00041

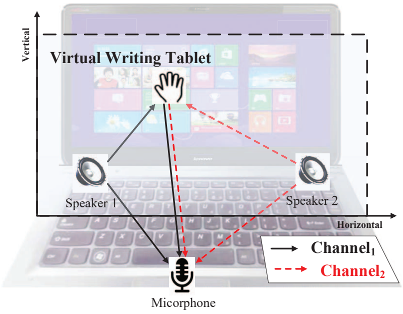


Fig. 1. Illustration of the virtual writing tablet on a laptop.

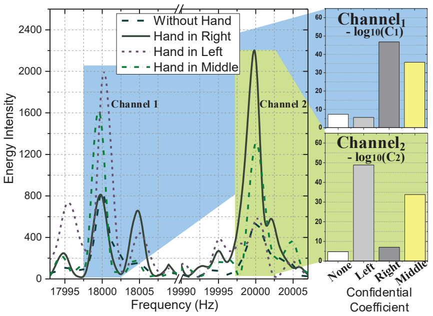


Fig. 2. Illustration of the confidential coefficients of the acoustic signals under different hand positions.

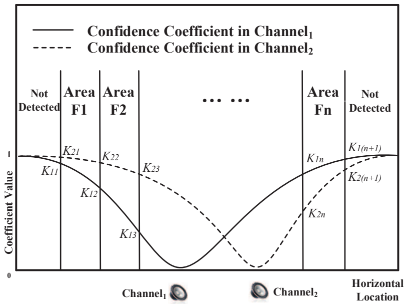


Fig. 3. Illustration of horizontal areas dividing.

ing: real-time and accuracy, VPad realizes hand movement tracking by letting the two speakers emit the acoustic signals with different frequencies, and the microphone records the acoustic signals reflected by the user's hand. Specifically, we first propose *Sliding-window Overlap Fourier Transformation* (*SOFT*) technique to increase measuring resolution for real-time tracking. The trajectory of each hand movement can be decomposed into horizontal and vertical movements. VPad first identifies the energy patterns of the reflected acoustic signals to continuously track the hand's horizontal movements, and then uses Doppler frequency shift of the acoustic signals to carry out tracking of the vertical movements. To realize the final step of character recognition, we use a stroke direction sequence model based on possibility estimation to deal with different writing habits and recognize the exact characters written in the air.

We highlight our main contributions as follows:

- We propose VPad to enable writing in the air for traditional commercial laptops leveraging the existing audio devices including two speakers and one microphone on a laptop.
- We utilize both frequency shift and energy feature to enable VPad accurately track hand movements, and present SOFT technique and use a non-rectangular window function for real-time tracking.
- We employ a stroke direction sequence model based on possibility estimation to recognize exact characters users write in the air, which handles different writing habits.
- Our experimental results with multiple participants show that the character recognition accuracy of VPad is higher than 90% in different environments, and the average error of trajectory tracking is $1.55cm$.

The rest of this paper is organized as follows. We analyze the feasibility of several fundamental techniques in Section II. Section III presents the system architecture and design details of VPad. We evaluate the performance of VPad and present the results in Section IV. Finally, we review the related work and give conclusive remarks in Section V and VI respectively.

II. Feasibility Study

In this section, we present the feasibilities of some potential techniques, including energy features and Doppler shifts, on the acoustic-based hand movement tracking with a single laptop, which serve as the foundation for our system.

A. Tracking the Horizontal Movement Velocity using Energy Features of Acoustic Signals

As illustrated in Fig. 1, the left and right speakers emit acoustic signals with different frequencies, which generates two transmitting channels, i.e., $Channel_1$ and $Channel_2$. When the hand is put on the top of keyboard, there are two transmitting paths for each channel. The energy[1] of received acoustic signal from each channel can be represented as $E = E_0 + E_1$, where E_0 denotes the energy of Line-Of-Sight (LOS) signals, and E_1 denotes the energy of acoustic signals reflected by the hand. Therefore, the energy of acoustic signals received by the microphone increases dramatically when a hand is above keyboard.

We assume E_0 obeys Gaussian distribution, and a sample set E' of E_0 obeys T-distribution. For an acoustic signal s received by the microphone, the confidential coefficient c of signal s relative to E_0 can be presented as

$$c = \int_{-\infty}^{t=-|t_0|} P(n,t) + \int_{t=|t_0|}^{\infty} P(n,t), \tag{1}$$

where $P(n,t)$ is the possibility distribution function, and

$$t_0 = \frac{1}{\sigma_{E'}}(\overline{E'} - E_s)\sqrt{n-1}, \tag{2}$$

where $\overline{E'}$ and $\sigma_{E'}$ are the mean and variance of E' respectively, E_s is the energy of signal s, and n is the size of sample set. From Eq. (2), we notice if E_s goes far away from the expectation of E_0, the value of t_0 would increase. Combine with Eq. (1), we find that both parts of the confidential coefficient, i.e., $\int_{-\infty}^{t=-|t_0|} P(n,t)$ and $\int_{t=|t_0|}^{\infty} P(n,t)$ would decrease, leading to the decrease of confidential coefficient c, i.e., the possibility that signal s is a sample of E_0 decreases and further a higher proportion of signal s' energy comes from acoustic signal reflected by the hand, vice versa.

Based on Eq. (1), an acoustic signal received by the microphone has a unique energy feature $< c_1, c_2 >$, where c_1

[1] The energy of acoustic signals here is defined as the amplitude of the acoustic signals in frequency domain.

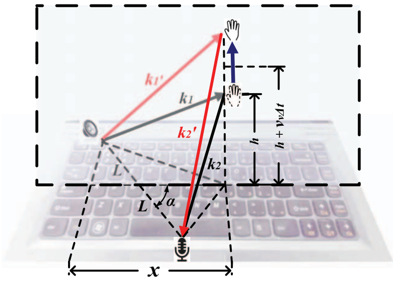


Fig. 4. Illustration of the vertical movement.

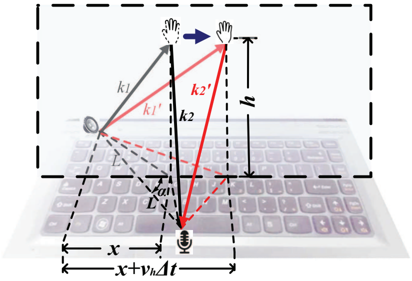


Fig. 5. Illustration of the horizontal movement.

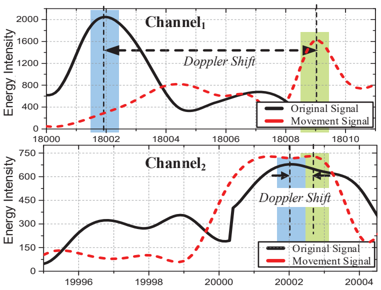


Fig. 6. Doppler frequency shift when the hand movement at the left of the virtual writing tablet.

and c_2 denote the confidential coefficients in Channel$_1$ and Channel$_2$ respectively. To capture the unique energy feature, we trace the acoustic signal received by the microphone under four conditions, i.e., without hand, hand in the right, hand in the left and hand in the middle, and calculate energy features $< c_1, c_2 >$ for each condition. Fig. 2 illustrates confidential coefficients of the acoustic signal under different hand positions. We can see that in each channel, the values of confidential coefficient present significant differences under different conditions. Thus, the energy feature $< c_1, c_2 >$ of acoustic signals received by the microphone is dominated by hand positions. To further utilize the energy feature in tracking hand position, we divide the 2-D virtual plane into n horizontal areas, i.e., F_1, ...F_n, as shown in Fig. 3. When the hand is in the area F_i, the extracted energy feature would have similar patterns, i.e.,

$$\begin{cases} c_1 \in [min(K_{1i}, K_{1(i+1)}), max(K_{1i}, K_{1(i+1)})] \\ c_2 \in [min(K_{2i}, K_{2(i+1)}), max(K_{2i}, K_{2(i+1)})], \end{cases} \tag{3}$$

where K_{1i} and K_{2i} are thresholds of F_i's confidence coefficient from $Channel_1$ and $Channel_2$ respectively. Therefore, we can track the horizontal hand position by comparing the patterns of received acoustic signal on energy feature with that of each area.

During the time period of Δt, if a user's hand moves from the area F_a to the area F_b, the horizontal movement velocity v_h can be obtained by

$$v_h = \frac{x_a - x_b}{\Delta t}, \tag{4}$$

where x_a and x_b are the horizontal positions of F_a and F_b's center points respectively.

B. Tracking the Vertical Movement Velocity using Doppler Shifts of Acoustic Signals

We also study the feasibility of utilizing energy features of acoustic signals to track the vertical movement. However, we find that tracking the vertical movement through energy features does not achieve acceptable results, so we adopt Doppler shift to track the vertical movement velocity. For each channel, during a hand movement, the propagating distance of acoustic signal reflected by the hand would change, which leads to *Doppler shift* [4]. The Doppler frequency shift, Δf, can be represented as $\Delta f = v f_0 / v_0$, where f_0 and v_0 are the frequency and speed of the signal respectively, and v is the rate of propagating distance's change.

Fig. 4 shows an example of hand vertical movement. The distance between speaker and microphone is $2L$, and $\angle\alpha$ denotes the angle between microphone-speaker line and horizontal line, both of which are known constants. Although the positions of microphones and speakers in different laptops vary with each other, the method is still effective as long as users provide the relative position information in advance.

For hand vertical movement from t_0 to $t_0 + \Delta t$, the propagating distance of acoustic signals reflected from the hand is $s_1 = k_1 + k_2 = \sqrt{x^2 + (L\sin\alpha)^2 + h^2} + \sqrt{(x - 2L\cos\alpha)^2 + (L\sin\alpha)^2 + h^2}$, where h denotes the hand vertical position at t_0 (i.e., the height of hand relative to the keyboard), x denotes the hand horizontal position at t_0 (i.e., the horizontal distance from the left speaker to the hand position). Let v_v denote the velocity of hand vertical movement. After one time unit Δt, the propagating distance of the acoustic signal is $s_2 = k_1' + k_2' = \sqrt{x^2 + (L\sin\alpha)^2 + (h + v_v\Delta t)^2} + \sqrt{(x - 2L\cos\alpha)^2 + (L\sin\alpha)^2 + (h + v_v\Delta t)^2}$. Based on the two distance, the rate of acoustic signal propagating distance change during Δt is

$$v_{pv} = \frac{\Delta s}{\Delta t} = \frac{d(|s_2 - s_1|)}{dt} = \frac{h v_v}{\sqrt{x^2 + (L\sin\alpha)^2 + h^2}} + \frac{h v_v}{\sqrt{(x - 2L\cos\alpha)^2 + (L\sin\alpha)^2 + h^2}}.$$

Therefore, Doppler frequency shift caused by the vertical movement is

$$\Delta f_1 = f_1(v_v, x, h) = \frac{v_{pv} f_0}{v_0}.$$

Similarly, for a hand *Horizontal Movement* from t_0 to $t_0 + \Delta t$, as shown in Fig. 5, the rate of acoustic signal propagating distance change during Δt can be represented as

$$v_{ph} = \frac{x v_h}{\sqrt{x^2 + (L\sin\alpha)^2 + h^2}} + \frac{x v_h}{\sqrt{(x - 2L\cos\alpha)^2 + (L\sin\alpha)^2 + h^2}},$$

where v_h denotes the velocity of the hand horizontal movement. Therefore, Doppler shift caused by the horizontal movement can be presented as

$$\Delta f_2 = f_2(v_h, x, h) = \frac{v_{ph} f_0}{v_0}.$$

Doppler shift Δf of the acoustic signal from one channel is the composition of Δf_1 and Δf_2, i.e.,

$$\Delta f = \sqrt{\Delta f_1^2 + \Delta f_2^2} = \sqrt{f_1^2(v_v, x, h) + f_2^2(v_h, x, h)}. \quad (5)$$

Theoretically, there are two Doppler shifts from Channel$_1$ and Channel$_2$ respectively. However, if the hand movement is at the left of virtual writing plane, Doppler shifts from Channel$_2$ (i.e., from the right speaker to the microphone) are too weak to measure, as shown in Fig. 6, vice versa. Hence, Doppler shift from Channel$_1$ and Channel$_2$ are used to track the hand movement at the left and right of the plane respectively. Assuming that x and h are known, and Doppler shift Δf from Channel$_1$ or Channel$_2$ is accurately measured. Since we have the horizontal movement velocity v_h in Δt based on energy features of acoustic signals, the vertical movement velocity v_v in Δt can be calculated based on Eq. (5).

III. System Design

From the above study, we know it is feasible to track users' hand movements for laptops leveraging acoustic signals. In this section, we present the design of our proposed system, VPad, which tracks the user's hand movement through energy features and Doppler shifts of acoustic signals propagating from two speakers to one microphone in a laptop.

A. System Overview

VPad uses acoustic signals to continuously track the hand movement trajectory, the design of emitted acoustic signals is thus critical. According to Doppler effect, with the same rate of propagating distance's change, the higher original frequency of the emitted acoustic signal leads to the larger frequency shift. Since most laptops only support the sampling rate up to $44.1kHz$, the highest sound frequency can be used is around $22kHz$. Thus, VPad generates acoustic signals with the frequency of $18kHz$ and $20kHz$ from two speakers respectively, which are inaudible to most people [10].

The workflow of VPad is shown in Fig. 7. In *Processing Acoustic Signal*, VPad transforms the time domain acoustic signals into frequency domain signals with the resolution of $1Hz$ for real-time tracking. In *Tracking Trajectory*, VPad decomposes each hand movement into horizontal and vertical movements. VPad first identifies energy patterns of reflected acoustic signals to continuously track hand horizontal movements, and then uses Doppler shift of acoustic signals for tracking of vertical movements. Combined with the estimation of initial position, VPad can track the trajectory of each hand movement. Finally, in *Recognizing Character*, VPad recognizes exact writing characters using a stroke direction sequence model based on possibility estimation.

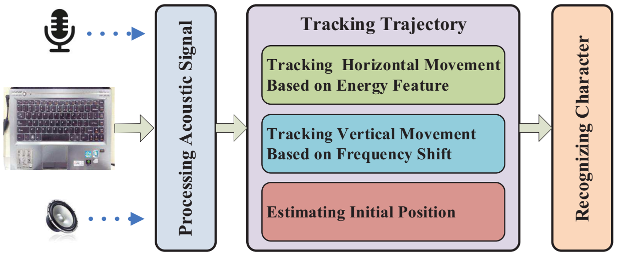

Processing Acoustic Signal
Tracking Trajectory
Tracking Horizontal Movement Based on Energy Feature
Tracking Vertical Movement Based on Frequency Shift
Estimating Initial Position
Recognizing Character

Fig. 7. The workflow of VPad

B. Processing Acoustic Signal

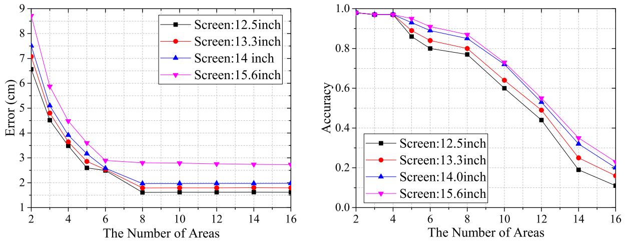
VPad transforms a time domain signal into a frequency domain signal using *Discrete Fourier Transformation* (*DFT*). For $20kHz$ and $18kHz$ acoustic signals, VPad need do DFT at least with 40000-point and 36000-point respectively to reach the frequency resolution of $1Hz$. Hence, VPad used 40000-point DFT to optimize signal processing. Because of the limitation of laptops' hardware, the sampling rate is less than $44.1kHz$, so the measure time interval is nearly $0.9s$ (i.e., $40000\ point/44100Hz$). However, the time interval of $0.9s$ is too long to detect the hand movement in real-time.

In order to improve time resolution, we present *Sliding-window Overlap Fourier Transformation*(*SOFT*) method, which uses a sliding window whose length is $0.9s$ with step $0.1s$. VPad performs DFT in each overlapped sliding window to improve time resolution from $0.9s$ to $0.1s$. Note that DFT algorithm requires the number of sampling points to be 2^n, otherwise it would result in the frequency domain signal distortion, i.e., *Fense Effect* [11]. We thus add zeros at the end of each sliding-window until the number of points achieves 2^{16} to eliminate Fense Effect. After each sampling step (i.e., $0.1s$), VPad only keeps the sampled data collected in the latest $0.9s$ and performs DFT. Through SOFT method, VPad is able to track the hand movement trajectory in real-time with only $0.1s$ time resolution. However, due to frequency leakage distortion [11], it is hard to accurately capture frequency shifts in real environments. The frequency leakage introduces spurious high-frequency components into the spectrum which declines the accuracy. VPad uses nonrectangular window, such as Hamming window [11], to suppress spurious high-frequency components.

C. Tracking Trajectory

VPad's trajectory tracking algorithm is built on the principle introduced in feasibility study. In this section, we describe the design of trajectory tracking in detail.

1) Tracking Horizontal Movements: VPad utilizes energy features to track horizontal hand movements. The virtual writing plane is divided into several areas for estimating the hand's horizontal position. If the user's hand is in one of divided areas, the hand horizontal position is approximately regarded as the horizontal position of the area's center point. Therefore, more divided areas could improve the estimation performance of horizontal movements. However, more divided



(a) Error.

(b) Accuracy.

Fig. 8. Performance of the horizontal position estimation.

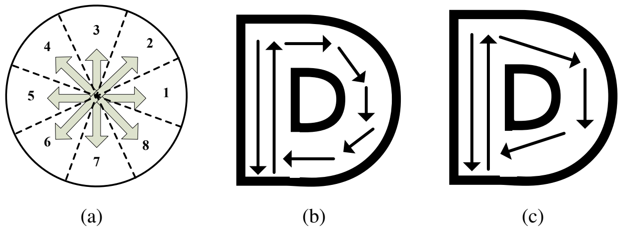
areas would also decrease the area estimation accuracy due to ambient noises or device fluctuations.

We study empirically the impact of the different number of areas on the horizontal position estimation. We recruit 20 volunteers (10 males and 10 females), and each volunteer is asked to put their hand on n ($n = 2 \cdots 16$) different positions at the virtual plane of VPad. We enable the front camera to record the actual horizontal position $\{x_1, x_2, ..., x_n\}$ of users as ground truths, and the horizontal position $\{x'_1, x'_2, ..., x'_n\}$ estimated by VPad as test samples. If x_i and x'_i belong to the same area, we regard it as a correct area estimation. Since most of laptops equip with 12.5-*inch*, 13.3-*inch*, 14.0-*inch* and 15.6-*inch* screens at present, we repeat the experiment on four types of laptops respectively. The results are shown in Fig. 8(a) and 8(b). The error is that the average distance difference in horizontal position between the ground truths and test samples. The area estimation accuracy is that the proportion of the correct area estimations in all area estimations.

From Fig. 8(a) and 8(b), we observe that 6-8 divided areas can achieve better performance on both error and accuracy. Also, we find that more areas no longer reduce the error, while result in more mistaken area estimations. From these observations, VPad employs 8 divided areas to estimate the horizontal movement velocity. After the virtual writing plane is divided into 8 horizontal areas, VPad can track the horizontal position of the hand by comparing patterns of current signal sample on energy feature with that of each area, and determines the horizontal movement velocity v_h based on Eq. (4).

2) Tracking Vertical Movements: VPad tracks vertical hand movement trajectories based on Doppler effect. VPad first extracts Doppler shift from received acoustic signals. Then, combined with the horizontal movement velocity, the vertical movement velocity v_v can be calculated based on Eq. (5).

3) Estimating Initial Position: Except for tracking horizontal and vertical movement velocities, VPad needs to estimate the initial horizontal and vertical positions before tracking hand trajectory. When the user's hand starts to move, VPad first compares energy features of received acoustic signal with theoretical energy features of each area to estimate the initial horizontal position x_0 at $t = 0$. Then VPad uses the time difference of arrival (TDoA) of two received acoustic signals (i.e., the signal propagating from the speaker to microphone directly and that reflected by the hand) to estimate the distance difference between two paths, and finally determines the initial vertical height h_0 at $t = 0$. To ensure the tracking trajectories are continuous, VPad sets the initial position of trajectory segment in time t, i.e., (x_t, h_t), as the ending position of the trajectory segment in time $t - 1$.



(a)

(b)

(c)

Fig. 9. Illustration of the trajectory recognition algorithm, (a) the eight movement directions in a plane; (b) and (c) two different stroke sequences of the character 'D'.

4) Tracking Trajectory: VPad resolves the hand velocity into the horizontal and vertical velocities. Based on horizontal and vertical movement tracking, VPad obtains the horizontal velocity v_h and vertical velocity v_v in Δt time. Using vector composition method, the two-dimensional movement velocity $\overline{v}$ in Δt is $\overline{v} = v_v \times \overline{i} + v_h \times \overline{j}$. Then, VPad can track the hand movement trajectory $\overline{s}$ via the integration of the velocity $\overline{v}$ from t_0 to $t_0 + \Delta t$, i.e.,

$$\overline{s} = \int_{t_0}^{t_0+\Delta t} \overline{v}. \tag{6}$$

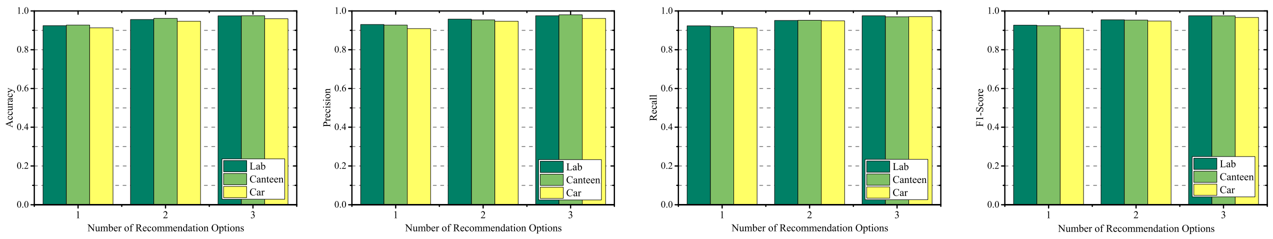
Finally, with the estimation of initial position, we can continuously track hand movement trajectories during any time.

D. Recognizing Character

In this section, we propose an algorithm for VPad to recognize exact character the user writes in the air. When the user writes a character on VPad, the writing trajectory can be converted into a stroke direction sequence. VPad compares the sequence with potential stroke direction sequences of all possible characters to find out the most similar one.

To ensure the robustness of character recognition, we first divide hand movement directions into eight basic directions, as shown in Fig. 9(a). Thus, VPad can transform a stroke direction sequence of characters into a number sequence. For example, one stroke direction sequence of the character 'D' can be regarded as a sequence $S = [7, 3, 1, 8, 7, 5, 4]$, as shown in Fig. 9(b). However, the stroke direction sequence varies from one user to another, so the character 'D' may be written as another sequence, e.g., $S = [7, 3, 8, 7, 5]$ in Fig. 9(c). We add all potential stroke direction sequences of a character to a list as $G_{char=C} = \{S_1, S_2, ...\}$, where C is a character such as 'A', 'D', etc, and the potential sequences of all characters can be defined as a set, i.e., $G\{G_{char='A'}, G_{char='B'}, ...\}$.

For matching stroke direction sequences, we use the Weighted Minimum Edit Distance (WMED) [12] to represent the similarity between two sequences. The minimum edit distance between two sequences is defined as the minimum number of operations (*insertion*, *deletion*, *substitution*) that



(a) Accuracy (b) Precision (c) Recall (d) F1 Measure

Fig. 10. Overall performance of VPad.

one sequence needs to be transformed into another. To improve the accuracy of proposed algorithm, we assign a weight for each *substitution* operation. If a stroke direction n_0 is substituted by another one n_1, the weight of substitution operation is represented as

$$w = \begin{cases} |n_0 - n_1| & \text{if } 1 <= |n_0 - n_1| <= 4 \\ 8 - |n_0 - n_1| & \text{if } 5 <= |n_0 - n_1| <= 7, \end{cases} \tag{7}$$

where w represents the similarity between two stroke directions n_0 and n_1. Thus, the value of WMED between two sequences is the sum of the weight values of all substitution operations and the number of insertion and deletion operations.

For a stroke direction sequence Q which is extracted from a trajectory the user writes in the air, VPad scans all sequences in the list G and chooses the k-Nearest-Sequences of Q which have the k least WMED values as a set $V = \{< S_1, m_1 >, < S_2, m_2 >, ..., < S_k, m_k >\}$, where S_i is the i^{th} sequences in the set V, and m_i is the value of WMED between S_i and Q. The $P_{_{char=C}}$, the possibility of the stroke direction sequences being corresponding to a special character C, is

$$P_{_{char=c}} = \frac{\sum_{S_k \in G_{char=c} \wedge < S_k, m_k > \in V} \frac{1}{m_k}}{\sum_{< S_k, m_k > \in V} \frac{1}{m_k}}. \tag{8}$$

Finally, VPad can recommend several character options based on the order of their possibility.

IV. Evaluation

In this section, we evaluate the performance of our system, VPad, with four traditional off-the-shelf laptops under three real environments.

A. Experimental Setup and Methodology

We use a Lenovo S230u with 12.5-inch screen, a Xiaomi Air with 13.3-inch screen, a Lenovo V470 with 14-inch screen and a Lenovo Y550 with 15.6-inch screen as experiment facilities. Only the built-in audio devices of laptops, one microphone and two speakers, are used in experiments.

We conduct experiments with 20 volunteers (10 males and 10 females). The number of volunteers with ages ranging from 20 to 40 are 14, and that with ages ranging from 41 to 65 are 6. 70 characters are tested to evaluate the performance of VPad including 26 capital letters (i.e.,'A'-'Z'), 26 lowercase letters (i.e., 'a'-'z'), 10 numbers (i.e., 0-9) and 8 special characters (i.e., Δ, Γ, Ω, Π, Σ, $\angle$, $\wedge$, $\vee$). The experiments are conducted in three real environments, i.e., a lab, a noisy canteen, and a moving car. In each environment, each user is asked to write all characters twice with VPad on four laptops, i.e., totally 280 writings for a user in each environment. Each user writes the characters in the air with his/her own writing habit, regardless of the writing speed, the size of writing character.

Several metrics are used in our evaluation. Assume i is the character users supposed to write, and j is the character recognized by VPad. Let ρ_{ij} denote the number of recognition results that recognize a character i as the character j.

Accuracy: The probability that an event is exactly identified for all type of events, i.e., $Accuracy = \sum_{i=1}^{n} \rho_{ii} / \sum_{j=1}^{n} \sum_{i=1}^{n} \rho_{ij}$.

Precision: The probability that the identification for an event A is exactly A in ground truth, i.e., $Precision_k = \rho_{kk} / \sum_{i=1}^{n} \rho_{ik}$.

Recall: The probability that an event A in ground truth is identified as A, i.e., $Recall_k = \rho_{kk} / \sum_{i=1}^{n} \rho_{ki}$.

F1-Score: A metric that combines precision and recall, i.e., $F1\_Score_k = 2 \times \frac{Precision_k \times Recall_k}{Precision_k + Recall_k}$.

B. Overall Performance

In each environment, the laptop's screen displays a trajectory tracked by VPad in real-time when a user writes a character in the air, and displays several character recommended options after writing. Note that all writings in the air follow each user's own writing habit, which are not always standard writings.

Fig. 10 shows the accuracy of VPad for each environment, and the average precision, recall, and F1-score value of all characters for each environment. It can be observed from the figure that the performances of VPad in different environments present insignificant differences. For one character recommendation option, the accuracies of VPad are all above 90% under three different environments. When VPad recommends three character options, the accuracy approaches 95% under three different environments. Meanwhile, F1-score of one, two and three character recommendation options are all above 0.9, 0.95 and 0.95 respectively under three different environments. This demonstrates VPad is insensitive to ambient influences, such as surrounding noises and vibrations.

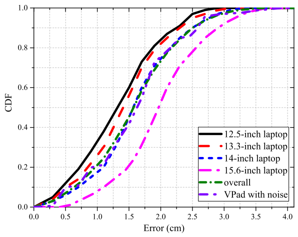


Fig. 11. CDF of trajectory tracking error under different laptops without and with noise.

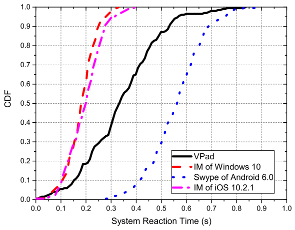


Fig. 12. CDF of the system reaction time for VPad.

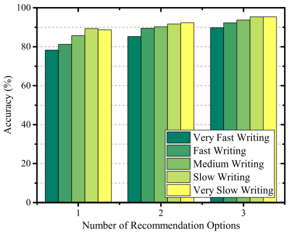


Fig. 13. Accuracy of VPad under different writing speeds.

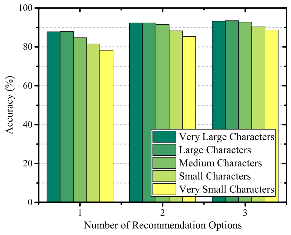


Fig. 14. Accuracy of VPad under different character sizes.

C. Performance of Trajectory Tracking

In this experiment, a user draws a line from an initial point to a target point in the air. For each point in the trajectory, we calculate the distance between the point and source-to-target connection line. The average distance of all points is regarded as the error of trajectory tracking. Each user conducts the experiment 20 times. Fig. 11 shows CDF of trajectory tracking error under four types of laptops. It can be seen that 80% of trajectory tracking errors are lower than $2.5cm$ under the four laptops. The average error of trajectory tracking is $1.55cm$. We also evaluate the robustness of VPad against background noise, in which users write characters in offices where people walk or talk around. Fig. 11 shows relative results. We can see that the average error of VPad with noise is almost the same with that without noise. This is because acoustic signal energy around the laptop is significantly higher than that elsewhere.

D. System Reaction Time of VPad

In this experiment, we ask the 20 volunteers to write all characters on devices with VPad and touch screens, i.e., a Lenovo S230u with VPad, a Lenovo S230u with the naive handwriting input method (IM) in Windows 10, a Huawei Honor X2 with the Huawei Swype IM deployed in Android 6.0, and a iPad Air 2 with its naive handwriting IM in iOS 10.2.1. And each character is written four rounds on four kinds of devices respectively. In each device, we deploy a screencast software to record the handwriting process. And for each writing, we enable the camera in each device to trace users' writing. Through analyzing the frames of the camera records (the frame rate is $25Hz$) and the frames of the screencasts (the frame rate is $30Hz$), we are able to get the ending time T_e that user writes a character in the air or touch screens and the time T_{dev} that the recognized character is displayed on screen. We define the system reaction time as $T = T_{dev} - T_e$.

Fig. 12 shows CDF of the system reaction time for the four devices. The average system reaction time of VPad, IM of Windows 10, Swype of Android 6.0 and IM of iOS 10.2.1 are $0.34s$, $0.19s$, $0.56s$ and $0.21s$ respectively. The system reaction time of IM in Windows 10 and iOS 10.2.1 are less because their handwriting IMs are tightly integrated into the OSes. Instead, VPad and Swype are both third-party softwares, which cannot fully utilize the capability of OSes. Although the system reaction time of VPad is larger than that of the naive IM of Windows 10, the average system reaction time difference between VPad and IM of Windows 10 is only $0.15s$, which is so little for users to be aware. Also, as third-party softwares, the system reaction time of Swype is $0.7s$ for 90% samples, but that of VPad is only $0.5s$. Therefore, VPad is able to achieve ideal performance, and meets users' actual demand as a virtual writing tablet.

E. Impact of Writing Speed in the Air

The speed of the user's writing in the air may possibly impact on the accuracy of VPad. We define the writing speed v_C as the writing trajectories' length $|\overline{s}|$ of a character C in the air divides the writing duration Δt, i.e., $v_C = |\overline{s}|/\Delta t$. We analyze all writing speeds of 20 users and find that the distribution of the writing speed satisfies the Gaussian distribution. Thus, 4 percentiles of the distribution (i.e., 0.05-percentile, 0.2-percentile, 0.8-percentile and 0.95-percentile) are exploited to divide the writing samples into 5 categories, i.e., very fast writing ($v_C > 100cm/s$), fast writing ($30cm/ms < v_C \leq 100cm/s$), medium writing ($10cm/s < v_C \leq 30cm/s$), slow writing ($v_C \leq 10cm/s$) and very slow writing ($v_C \leq 5.5cm/ms$).

Fig. 13 shows the accuracy of VPad under different writing speeds. We can see that the accuracy increases as the writing speed decrease from fast to slow. But the differences of accuracies between very fast writing and very slow writing are only 10.42%, 7.08% and 5.08% under one, two and three recommendation options respectively. This result shows that VPad is not sensitive to the writing speed of users.

F. Impact of Writing Character's Size in the Air

The size of writing character in the air may possibly have impact on the accuracy of VPad. For each writing character C, we use a rectangle to surround C, and the area of the smallest rectangle $S_{char=C}$ is called as the absolute size of C. To eliminate the impact of virtual writing tablet's size (i.e., laptops' screen size), we transform the absolute size of character to the character proportion P, which is defined as the proportion of surrounded rectangle area $S_{char=C}$ among the laptop's screen area $S_{laptop=L}$, i.e., $P = S_{char=C}/S_{laptop=L}$. We analyze all writing characters' sizes of 20 users and find that the distribution of characters' sizes satisfies the Gaussian distribution. Thus, 4 percentiles of the distribution (i.e., 0.05-percentile, 0.2-percentile, 0.8-percentile and 0.95-percentile) are exploited to divide writing samples into 5 categories, i.e., very large characters ($P > 0.7$), large characters ($0.5 < P \leq$

0.7), medium characters ($0.3 < P \leq 0.5$), small characters ($0.15 < P \leq 0.3$) and very small characters ($P \leq 0.15$).

Fig. 14 shows the accuracy of VPad under different character sizes. Although the accuracy decreases as the character size decreases from very large to very small, accuracy degradations are only 9.50%, 7.42% and 4.52% under one, two and three recommendation options respectively. This result shows that VPad is insensitive to the size of writing characters.

V. Related Work

Existing works on hand movement tracking and handwriting recognition can be categorized as follows.

Motion sensor-based Tracking: Some works use motion sensors in mobile devices to capture human movements. For instance, [13] demonstrates that smart watches can track user's arm motions to know the typing content on a keyboard. However, all these systems require an external device such as a smartphone or a wearable.

Acoustic signal-based Tracking: Some existing works rely on acoustic signals, such as utilizing acoustic signal to snoop keystroke [14], [15], monitor human's sleep apnea situation [16], and indoor localization [17]. Regarding the motion tracking, SoundWave [4] first uses Doppler effect of acoustic signals to recognize gestures, which can only provide predefined gesture recognition. CAT [6] realizes a high precision tracker using acoustic signals. But CAT can only track hand movement through an additional acoustic-signal emitter from a smartphone which is held in the user's hand. More recently, LLAP [7] and FingerIO [8] design trajectory tracking algorithms for mobile devices to track users' fingers near the devices, and Strata [9] develops a fine-grained acoustic-based tracker for smartphones using the channel impulse response of acoustic signals. Due to different deployment of audio components in laptops and mobile devices, all these works cannot be adopted in laptops. Moreover, LLAP and FingerIO employ CW signals and OFDM pulses to achieve accurate tracking respectively, which are susceptible to the interference of background for trajectory tracking in a large distance.

Handwriting Recognition: Handwriting recognition has been widely studied in the past decades. Except for study on handwriting recognition on specific devices [18], most recent studies focus on offline handwriting recognition, i.e., recognizing a character or word after users' writing [19]. However, these approaches can only recognize characters in handwriting images. Since tracked hand movements are stroke direction sequences instead of images, these approaches cannot be adopted to recognize characters after users write in the air.

Unlike previous work, VPad only utilizes two speakers and one microphone on most commercial laptops, for tracking users' hand movement trajectories in the air without additional infrastructures, and adopts a light-weight method to recognize the handwriting characters in real time.

VI. Conclusions

In this paper, we design a virtual writing tablet for laptops based on acoustic signals, VPad, which can accurately track hand movements and recognize characters written in the air. Unlike existing works, we only use the built-in audio devices to realize VPad. First, to achieve high tracking accuracy, we present Sliding-window Overlapping Fourier Transformation technique to find Doppler shift with higher resolution in real time. Then, we propose a trajectory tracking algorithm based on frequency shifts and energy features of acoustic signals to track the user's hand movement. Finally, a stroke direction sequence model based on possibility estimation is employed to achieve exact character recognition. Extensive experiments verify the feasibility and effectiveness of VPad.

Acknowledgment

This research is sponsored by National Natural Science Foundation of China (No. 61772338). This work is also supported by China Scholarship Council.

References

[1] "Touchscreens in mobiledevices market:global industry analysis and forecast." http://www.persistencemarketresearch.com/, 2017.

[2] U. Lowell, "Multi-touch screen helps the disabled," https://www.uml.edu/News/stories/2010-11/student touch screen.aspx, 2011.

[3] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with microsoft kinect sensor: A review," *IEEE Transactions on Cybernetics*, vol. 43, no. 5, pp. 1318–1334, 2013.

[4] S. Gupta, D. Morris, S. Patel, and D. Tan, "Soundwave: using the doppler effect to sense gestures," in *Proc. ACM CHI'12*, Austin, Texas, 2012.

[5] S. Yun, Y. C. Chen, and L. Qiu, "Turning a mobile device into a mouse in the air," in *Proc. ACM MobiSys'15*, Florence, Italy, 2015.

[6] W. Mao, J. He, and L. Qiu, "Cat: High-precision acoustic motion tracking," in *Proc. ACM Mobicom'16*, New York, NY, USA, 2016.

[7] W. Wang, A. X. Liu, and K. Sun, "Device-free gesture tracking using acoustic signals," in *Proc. ACM Mobicom'16*, New York, NY, 2016.

[8] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota, "Fingerio: Using active sonar for fine-grained finger tracking," in *Proc. ACM CHI'16*, San Jose, CA, USA, 2016.

[9] S. Yun, Y.-C. Chen, H. Zheng, L. Qiu, and W. Mao, "Strata: Fine-Grained Acoustic-based Device-Free Tracking," in *Proc. ACM Mobisys'17*, Niagara Falls, NY, USA, 2017.

[10] P. G. Kannan, S. P. Venkatagiri, M. C. Chan, A. L. Ananda, and L. S. Peh, "Low cost crowd counting using audio tones," in *Proc. ACM Sensys'12*, Toronto, ON, Canada, 2012.

[11] A. V. Oppenheim and R. W. Schafer, "Discrete-time signal processing," *Prentice Hall Signal Processing*, vol. 23, no. 2, pp. 1–39, 1999.

[12] P. S. Deshpande, L. Malik, and S. Arora, "Fine classification & recognition of hand written devnagari characters with regular expressions & minimum edit distance method," *Journal of Computers*, vol. 3, no. 5, pp. 11–17, 2008.

[13] H. Wang, T. T. Lai, and R. R. Choudhury, "Mole:motion leaks through smartwatch sensors," in *Proc. ACM Mobicom'15*, Paris, France, 2015.

[14] J. Liu, Y. Wang, G. Kar, Y. Chen, J. Yang, and M. Gruteser, "Snooping keystrokes with mm-level audio ranging on a single phone," in *Proc. ACM Mobicom'15*, Paris, France, 2015.

[15] J. Wang, K. Zhao, X. Zhang, and et al., "Ubiquitous keyboard for small mobile devices: Harnessing multipath fading for fine-grained keystroke localization," in *Proc. ACM Mobisys'14*, Bretton Woods, NH, 2014.

[16] R. Nandakumar and et al., "Contactless sleep apnea detection on smartphones," in *Proc. ACM Mobisys'15*, Florence, Italy, 2015.

[17] W. Huang, Y. Xiong, X. Y. Li, H. Lin, and et al., "Shake and walk: Acoustic direction finding and fine-grained indoor localization using smartphones," in *Proc. IEEE INFOCOM'14*, Toronto, Canada, 2014.

[18] F. Nouboud and R. Plamondon, "On-line recognition of handprinted characters: Survey and beta tests," *Pattern Recognition*, vol. 23, no. 9, pp. 1031 – 1044, 1990.

[19] A. Graves and J. Schmidhuber, "Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks," in *Advances in Neural Information Processing Systems 21*, 2009, pp. 545–552.