Advance Access publication on
doi: 10.1093/comjnl/bxz153

Enable Traditional Laptops with Virtual Writing Capability Leveraging Acoustic Signals

Li Lu[1,*], Jian Liu[2], Jiadi Yu[1], Yingying Chen[2], Yanmin Zhu[1], Linghe Kong[1] and Minglu Li[1]

[1]*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China*
[2]*WINLAB and Department Electrical and Computer Engineering, Rutgers University, New Brunswick, NJ, USA*
**Corresponding author: luli_jtu@sjtu.edu.cn*

Human–computer interaction through touch screens plays an increasingly important role in our daily lives. Besides smartphones and tablets, laptops are the most prevalent mobile devices for both work and leisure. To satisfy the requirements of some applications, it is desirable to re-equip a typical laptop with both handwriting and drawing capability. 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, to accurately track hand movements and recognize writing characters in the air without additional hardware. Specifically, VPad emits inaudible acoustic signals from two speakers in a laptop and then analyzes energy features and Doppler shifts of acoustic signals received by the microphone to track the trajectory of hand movements. Furthermore, we propose a state machine-based trajectory optimization method to correct the unexpected trajectory and employ a stroke direction sequence model based on probability estimation to recognize characters users write in the air. Experimental results show that VPad achieves the average error of 1.55 *cm* for trajectory tracking and the accuracy over 90% of character recognition merely through built-in audio devices on a laptop.

Keywords: acoustic signals; laptops; virtual writing; trajectory tracking.

Received 9 October 2018; Revised 24 September 2019; Editorial Decision 17 November 2019
Handling editor: Suchi Bhandarkar

1. INTRODUCTION

Recently, an increasing number of applications require interactions between users and devices through touch screens. One report suggests that over 95% of smart devices are equipped with a touch screen [1]. This trend even spreads into traditional laptops that are the most popular mobile devices in both work and leisure besides smartphones and tablets. However, most traditional laptops are not equipped with touch screens. Although small touchpads on laptops provide the scrolling and swiping functions, the touchpads still cannot support the writing and drawing capabilities that serves as the basis of many new applications, such as Drawboard PDF [2] and WRITEit [3]. Several situations may hinder keyboard input with a laptop. For example, because of the vibration in a vehicle (e.g. cars, buses, airplanes), users cannot conveniently input with the keyboard. And for a variety of reasons, keyboard input will remain difficult for many people [4], calling for more accessible interaction, in keeping with augmented and virtual reality. Some existing works (including mature products and research studies) implement the trajectory tracking systems, which are summarized in Table 1. Toward this end, our goal is to design a system, which can accurately track trajectory in real time leveraging the common available audio infrastructures including a microphone and two speakers on the traditional laptops.

In this work, we take one step forward to develop a device-free virtual writing tablet (*VPad*) leveraging common available audio devices on traditional laptops without any additional hardware. With acoustic signals emitted from the laptop, VPad seeks to achieve the fine-grained trajectory tracking and

TABLE 1. Comparison between existing works on the trajectory tracking.

Work	Sensor	Strength	Weakness
LeapMotion [5]	Camera	High precision	Sensitivity to ambient light
Kinect [6]	Camera	High precision	Sensitivity to ambient light
Wang *et al.* [7]	RFID	Low-cost	Require coverage of RFID
Sun *et al.* [8]	WiFi	Low-cost	Require coverage of WiFi
SoundWave [9]	Acoustic	Device-free	Low precision
AAmouse [10]	Acoustic	High precision	Require holding devices
CAT [11]	Acoustic	High precision	Require holding devices
LLAP [12]	Acoustic	Device-free & High precision	Incapable for laptop
FingerIO [13]	Acoustic	Device-free & High precision	Incapable for laptop
Strata [14]	Acoustic	Device-free & High precision	Incapable for laptop

accurate character recognition. To enable the in-air virtual writing capability leveraging acoustic signals, we face several challenges in practice. First, the sampling rate is limited by laptops' audio hardware, which leads to the limited tracking resolution embedded in received acoustic signals. Second, 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.

VPad realizes hand movement tracking by emitting acoustic signals with different frequencies through the two speakers and recording signals reflected by the user's hand with the microphone. The trajectory of each hand movement can be decomposed into horizontal and vertical movements. VPad first identifies the energy patterns of reflected acoustic signals to continuously track the hand's horizontal movements and then uses Doppler shift of acoustic signals to track the vertical movements. We further develop the *trajectory tracking optimization* mechanism based on state machine to correct the unexpected trajectories from the tremble of hand movements or the vibration of user context (e.g. transportation vehicles). To realize the character recognition, we use a stroke direction sequence model based on probability estimation to deal with different writing habits and recognize the exact characters written in the air. Moreover, to achieve two critical factors in hand movement tracking, i.e. real time and accuracy, we develop an algorithm called *real-time frequency shift finding*, which first utilizes a *sliding-window overlap Fourier transformation* (*SOFT*) technique and non-rectangular window function to increase measuring resolution for real-time tracking and then employs the *weighted average frequency peak* to accurately estimate the Doppler shift in real time.

We highlight our contributions as follows.

- We propose VPad to enable in-air writing capability for traditional commercial laptops leveraging common available audio devices including two speakers and one microphone on a laptop.
- We utilize both energy feature and Doppler shift to enable VPad accurately track hand movements and apply SOFT technique and a non-rectangular window function to improve the resolution for real-time tracking.
- We develop the state machine approach to eliminate the unexpected trajectories so that VPad is robust to jitters in trajectory tracking.
- We employ a stroke direction sequence model based on probability estimation to recognize exact characters users write in the air, which handles different users' writing habits.
- Our experimental results with multiple participants show that the character recognition accuracy of VPad is higher than 90% under different environments, and the average error of trajectory tracking is 1.55 *cm*.

The rest of this paper is organized as follows. We investigate the feasibility of several fundamental techniques in Section 2. Section 3 presents the system architecture and design details of VPad. We show the implementation details of VPad in Section 4. We evaluate the performance of VPad and present the results in Section 5. Finally, we review the related work and make a conclusion in Sections 7 and 8, respectively.

2. FEASIBILITY STUDY

In this section, we investigate the feasibility of utilizing energy features and Doppler effect on the acoustic-based hand movement tracking with a single laptop, which serve as the foundation for our system. Figure 1 illustrates the virtual writing tablet for a laptop, which is a 2D virtual plane perpendicular to the keyboard. On the 2D plane, an arbitrary hand movement is a trajectory under a specific moving velocity at a unit time period. The hand movement velocity can be decomposed into the horizontal movement velocity and vertical movement velocity. In order to obtain these two velocities, we first estimate the horizontal movement velocity based on the energy features of the acoustic signals and then combine the horizontal movement velocity with Doppler shift to estimate the vertical movement velocity.

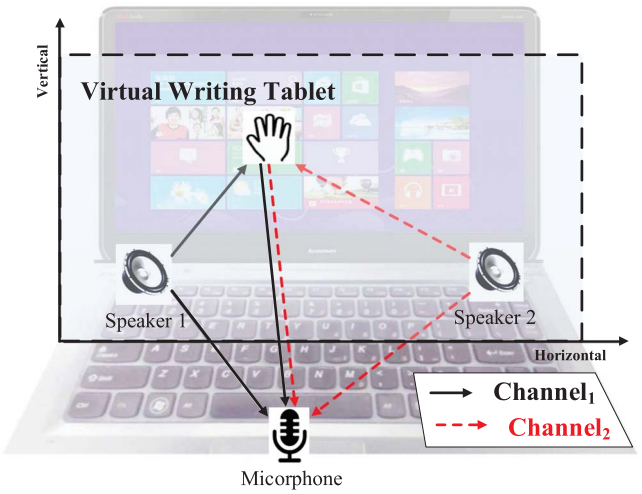


FIGURE 1. Illustration of the virtual writing tablet on a laptop.

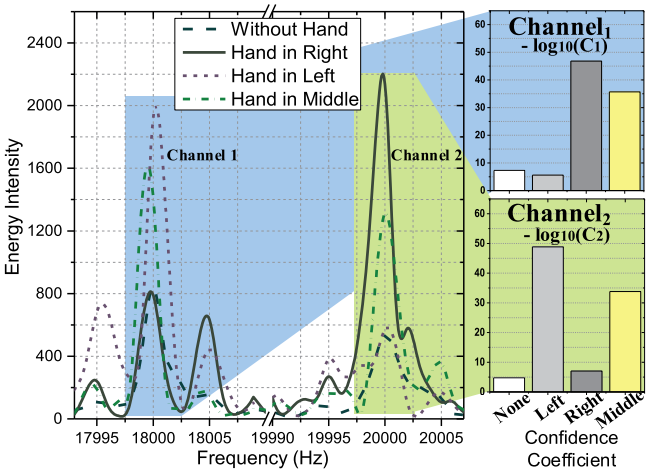


FIGURE 2. Illustration of the energy intensity and confidence coefficient of the acoustic signals.

2.1. Tracking the horizontal movement velocity using energy features of acoustic signals

To track the hand movement with acoustic signals, it is straightforward to consider utilizing the energy of acoustic signals. Specifically, during the propagation of acoustic signals, the energy of the signals can exhibit the propagation channel state. When a hand appears in the propagation channel of acoustic signals, the signals are absorbed, reflected or diffracted by the hand. This provides the opportunity to utilize the energy of acoustic signals to track the hand movement. As illustrated in Fig. 1, the left and the right speakers emit acoustic signals with different frequencies, which are regarded as two channels, i.e. $Channel_1$ and $Channel_2$. When the hand is put on the top of the keyboard, there are two dominant transmitting paths for each channel. The energy of received acoustic signal from each channel is:

$$E = E_0 + E_1, \tag{1}$$

where E_0 and E_1 are the energy of acoustic signals directly propagating from a speaker to the microphone and that reflected by the hand, respectively. Therefore, the energy of acoustic signals received by the microphone increases dramatically when a hand is above the keyboard.

Note that E_0 is a constant value in an ideal environment, regardless of the ambient noise and unstable audio recording hardware. We assume E_0 obeys Gaussian distribution, and thus a sample set E' of E_0 obeys T-distribution. For an acoustic signal s received by the microphone, the corresponding confidence coefficient c for E_0 is

$$c = \int_{-\infty}^{t=-|t_0|} P(n,t) + \int_{t=|t_0|}^{\infty} P(n,t), \tag{2}$$

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

$$t_0 = \frac{1}{\sigma_{E'}}(\overline{E'} - E_s)\sqrt{n-1}, \tag{3}$$

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 E'. From Equation (2) and (3), we find if E_s is significantly different from E_0, the value of t_0 would increase, which leads to the decrease of both parts of the confidence coefficient, i.e. $\int_{-\infty}^{t=-|t_0|} P(n,t)$ and $\int_{t=|t_0|}^{\infty} P(n,t)$. Hence, the confidence coefficient c would decrease, i.e. the probability that the signal s is a sample of E_0 decreases. This indicates the signal s is more probable to be induced by the hand when c decreases.

Based on confidence coefficients, an acoustic signal received by the microphone has a unique energy feature $< c_1, c_2 >$ based on Equation (2), where c_1 and c_2 are the confidence coefficients in $Channel_1$ and $Channel_2$, respectively. Figure 2 illustrates the energy intensity and confidence coefficient of the acoustic signal under different hand positions. We record the acoustic signals with the microphone under four conditions, i.e. without hand, hand in the right, hand in the left and hand in the middle and derive confidence coefficients $< c_1, c_2 >$ of received acoustic signals for each condition. We can see from Fig. 2 that in each channel, both the energy intensities and confidence coefficient values exhibit significant differences under different conditions. Thus, the energy feature $< c_1, c_2 >$ of the acoustic signal received by the microphone is dominated by the user's hand positions, which can be used to track the hand position.

We divide the 2D virtual plane into n horizontal areas, i.e. $F_1, F_2, \ldots F_n$, as shown in Fig. 3. When the hand is in the area

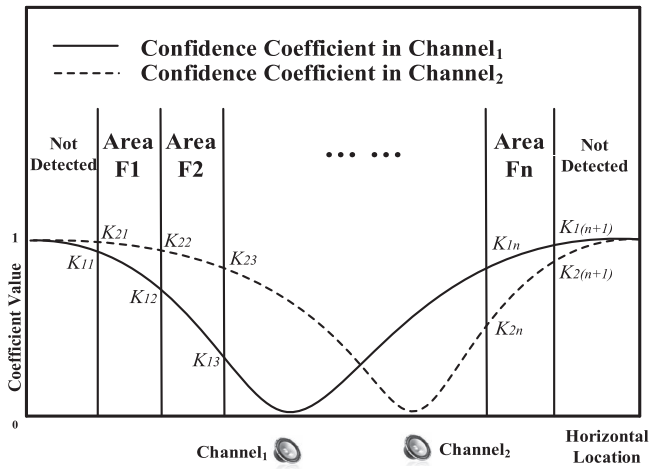


FIGURE 3. Illustration of horizontal areas dividing.

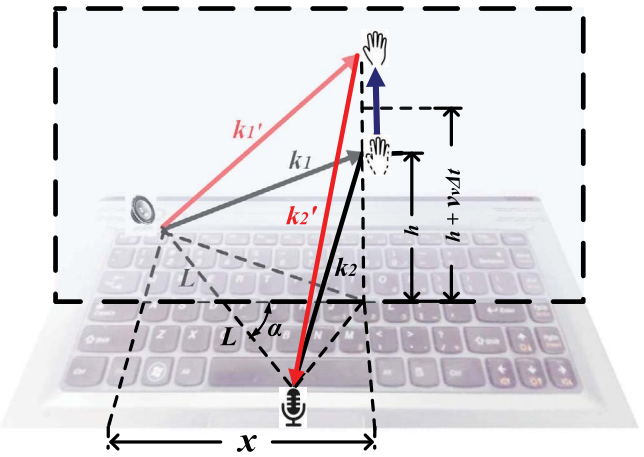


FIGURE 4. Illustration of the vertical movement.

F_i, the energy feature has similar patterns, i.e.

$$\begin{cases} c_1 \in [\min(K_{1i}, K_{1(i+1)}), \max(K_{1i}, K_{1(i+1)})] \\ c_2 \in [\min(K_{2i}, K_{2(i+1)}), \max(K_{2i}, K_{2(i+1)})], \end{cases} \tag{4}$$

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 position of the user's hand by identifying the energy features with the profiles in 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 approximately obtained by

$$v_h = \frac{x_a - x_b}{\Delta t}, \tag{5}$$

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

2.2. 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 with energy features does not achieve acceptable results, due to the shorter height in the vertical direction. Hence, we adopt Doppler shift to track the vertical movement velocity. For each channel, during a hand movement, the propagation distance of acoustic signal reflected by the hand changes, which induces *Doppler shift* [8]. Specifically, the Doppler shift Δf is

$$\Delta f = \frac{v f_0}{v_0}, \tag{6}$$

where f_0 and v_0 are the frequency and speed of the emitted signal respectively, and v is the change rate of propagation distance.

Based on Doppler shift, we can track the vertical hand movement. Figure 4 illustrates a vertical hand movement. The distance between speaker and microphone is $2L$, and the angle between microphone-speaker connection and horizontal line is $\angle\alpha$, both of which are already known. Although positions of microphones and speakers in different laptops vary with each other, the method is still efficient as long as users provide the relative position information in advance.

For vertical hand movement from t_0 to $t_0 + \Delta t$, the propagation 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 is the vertical hand position at t_0 (i.e. the height of hand relative to the keyboard), x is the hand horizontal position at t_0 (i.e. the horizontal distance from the left speaker to the hand position). After the time Δt, the propagation distance of acoustic signal is

$$s_2 = k_1' + k_2' = \sqrt{x^2 + (L\sin\alpha)^2 + (h + v_v\Delta t)^2} + \sqrt{(x - 2L\cos\alpha)^2 + (L\sin\alpha)^2 + (h + v_v\Delta t)^2}.$$

The change rate of signal propagation distance during Δt is

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

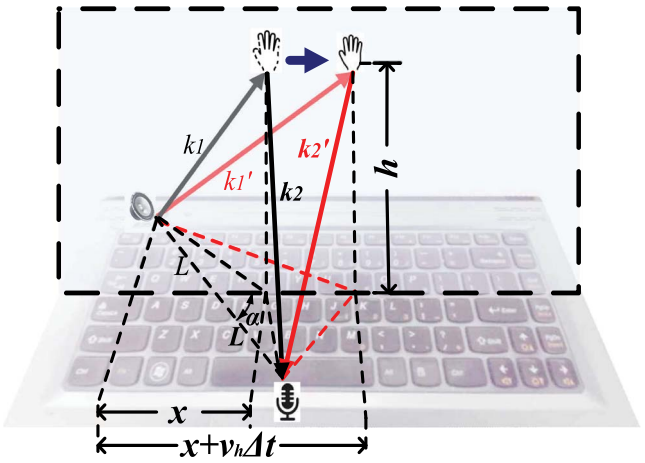


FIGURE 5. Illustration of the horizontal movement.

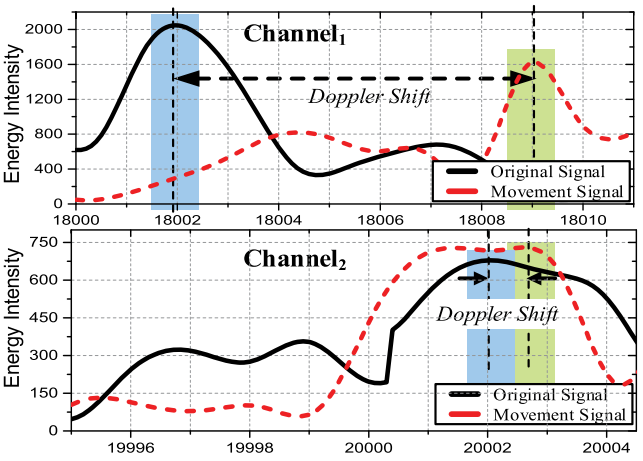


FIGURE 6. Doppler shift when the hand movement at the left of the virtual writing tablet.

Therefore, Doppler shift caused by the vertical movement is

$$\Delta f_1 = f_1(v_v, x, h) = \frac{v_{pv} f_0}{v_0}. \tag{7}$$

Similarly, for hand horizontal movement from t_0 to $t_0+\Delta t$, as shown in Fig. 5, the change rate of signal propagation distance during Δt is

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

where v_h is the velocity of hand horizontal movement. Therefore, Doppler shift caused by the horizontal movement is

$$\Delta f_2 = f_2(v_h, x, h) = \frac{v_{ph} f_0}{v_0}. \tag{8}$$

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

$$\Delta f = \sqrt{\Delta f_1^2 + \Delta f_2^2} = \sqrt{f_1^2(v_v, x, h) + f_2^2(v_h, x, h)}. \tag{9}$$

Theoretically, there are two Doppler shifts from Channel_1 and Channel_2, respectively. However, if the hand movement at the left of virtual writing plane, Doppler shifts of acoustic signals 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, 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 Equation (9).

3. SYSTEM DESIGN

Through the energy features and Doppler shifts of acoustic signals, a traditional laptop can track the trajectory of hand movements merely utilizing the built-in audio devices. In this section, we present the design of VPad, which tracks a user's hand movement and recognizes the writing characters in the air.

3.1. System overview

VPad emits the acoustic signals by the two speakers and receives the reflected signals by the microphone to track the hand movements. Since higher frequency of acoustic signals leads to more fine-grained velocity tracking based on Doppler effect, the frequency of emitted signals should be selected as large as possible. Most laptops only support the sampling rate up to 44.1 *kHz*, which induces the highest acoustic frequency of around 22 *kHz*. Thus, VPad emits acoustic signals with the frequency of 18 *kHz* and 20 *kHz* from two speakers respectively, which are inaudible to most people [15].

The workflow of VPad is shown in Fig. 7. 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. Then in *optimizing trajectory*, VPad further optimizes tracked trajectories to correct unexpected jitters. Finally, in *recognizing character*, VPad recognizes exact writing characters using a stroke direction sequence model based on probability estimation.

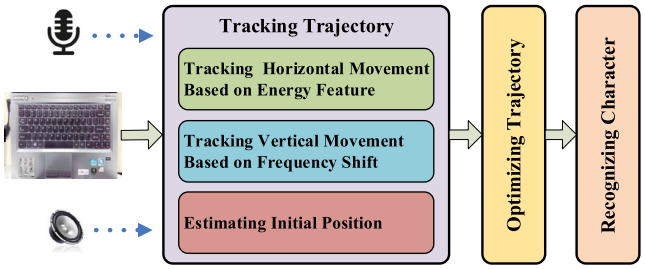


FIGURE 7. Workflow of VPad.

3.2. Tracking trajectory

VPad tracks the hand movement based on the principle as mentioned in Section 2. In this section, we describe the design of trajectory tracking in detail.

3.2.1. *Tracking horizontal movements*

VPad utilizes energy features to track horizontal hand movement trajectories. Before obtaining the energy feature, VPad first divides the virtual writing plane into several areas to estimate 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 movement velocity. On the other hand, more divided areas would decrease the estimation accuracy due to the ambient noise and device fluctuation.

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. 8a and b. The error is that the average distance difference of 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.

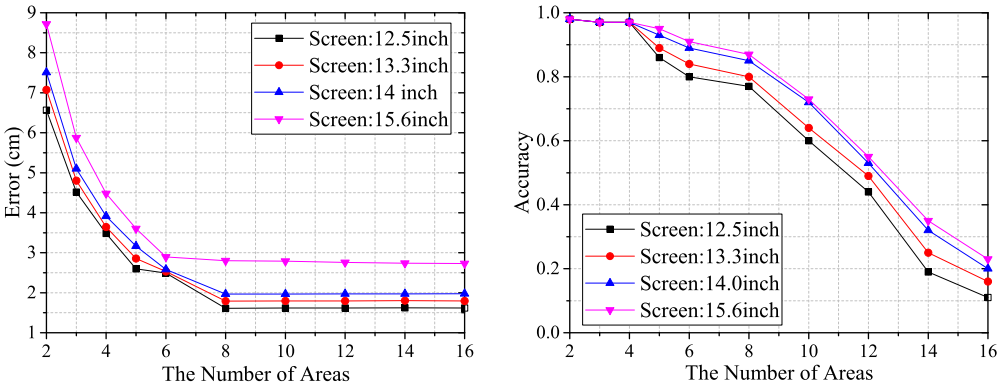
From Fig. 8a and b, we observe that six–eight divided areas can achieve better performance on both error and accuracy. Also, we find that more areas no longer reduce the error, while result in more mistaken area estimations. From these observations, VPad employs eight divided areas to estimate the horizontal movement velocity. After the virtual writing plane is divided into eight 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 Equation (5).

3.2.2. *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 Equation (9), i.e. $v_v = \sqrt{\Delta f^2 - f_2^2(v_h, x, h)} \cdot v_0/f_0$, where v_0 and f_0 are the speed and frequency of the emitted acoustic signal, respectively.

3.2.3. *Estimating initial position*

Except for tracking horizontal and vertical movement velocities, VPad needs to estimate the initial position before tracking hand trajectory. When the user's hand starts to move, VPad



(a) Error. (b) Accuracy.

FIGURE 8. Performance of the horizontal position estimation.

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) between line-of-sight (LOS) signal and reflected signal by 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 $t-1$.

Algorithm 1 Tracking Trajectory

Input: E_s: energy of received acoustic signal
Δf: Doppler shift of received acoustic signal
N: the number of horizontal areas
$< \overline{c}_1^i, \overline{c}_2^i >$: the profile of i^{th} horizontal area
ϵ: the threshold for horizontal area detection
INITIAL: the indicator for initial movement
Output: $\overline{s}$: hand movement trajectory

1: **if** INITIAL **then**
2: Derive the initial position (x_0, h_0) based on TDoA between LOS signal and reflected signal by hand.
3: **end if**
4: $x_{t-1} = x_0$, $h_{t-1} = h_0$.
5: $\overline{s} = NULL$
6: **while** $\Delta f \neq 0$ **do**
7: Derive the confidence coefficients c_1 and c_2 based on the energy E_s of received acoustic signals through Eq. (2).
8: **for** each $i \in [1, N]$ **do**
9: **if** $| < c_1, c_2 > - < \overline{c}_1^i, \overline{c}_2^i > | < \epsilon$ **then**
10: $x_t =$ the middle position of i^{th} area.
11: **break**.
12: **end if**
13: **end for**
14: $v_h = \frac{x_t - x_{t-1}}{\Delta t}$, where Δt is the unit time period between $t-1$ and t.
15: With the Doppler shift Δf, calculate the vertical velocity v_v of hand movement based on Eq. (9).
16: $\overline{v} = v_v \times \overline{i} + v_h \times \overline{j}$.
17: $s = \int_{t_0}^{t_0+\Delta t} \overline{v}$.
18: $\overline{s} = [\overline{s}, s]$.
19: $x_{t-1} = x_t$, $h_{t-1} = h_t$.
20: **end while**
21: **return** $\overline{s}$

3.2.4. *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 2D movement velocity $\overline{v}$ in Δt is:

$$\overline{v} = v_v \times \overline{i} + v_h \times \overline{j}. \tag{10}$$

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.

$$\overline{s} = \int_{t_0}^{t_0+\Delta t} \overline{v}. \tag{11}$$

Finally, with the estimation of initial position, we can continuously track hand movement trajectories during any time. Algorithm 1 shows the tracking trajectory of VPad.

3.3. Optimizing trajectory

Through above algorithms, VPad can track the user's hand movement during any time. However, the original trajectories may be unexpected ones. For instance, if the user draws a straight line with a slightly trembled hand, the trajectory detected by VPad is like a sawtooth. Moreover, in many transportation vehicles, such as cars, buses, ships and airplanes, even users' hands do not make jitters, the vibration of vehicles could lead to unexpected trajectories. In order to optimize the original trajectory, we propose a state machine-based algorithm to make it smooth.

We observe that, whether the trembled hand or vibration of transportation vehicles, the drift of trajectories is usually not significant compared with normal trajectories, and the trajectory direction usually changes but comes back soon to the previous direction when an unexpected jitter appears. Compared with unexpected situations, the normal trajectory direction usually keeps a relatively long time period without coming back to the previous direction.

Our trajectory optimization is based on the horizontal-direction and vertical-direction state machines. Figure 9 shows an example of tracking the handwriting of 'M' in the air based on the state machines. A horizontal-direction state machine has two states, of which the *left* state presents the sampled movement trajectory from right to left, and the *right* state presents the sampled movement trajectory from left to right, as shown in Fig. 10. When the state machine is in the left state, if the sampled movement trajectory is still from right to left, then the counter is zero; if the sampled movement trajectory is from left to right, then VPad adds one to the counter. Once the counter's value achieves a threshold N, the state transformation is triggered, i.e. a normal movement trajectory direction change is detected. In contrast, if the sampled movement trajectory direction recovers back to the previous direction before the value of counter reaches N, VPad regards the movement as a jitter. The vertical-direction state machine can be designed with the same method.

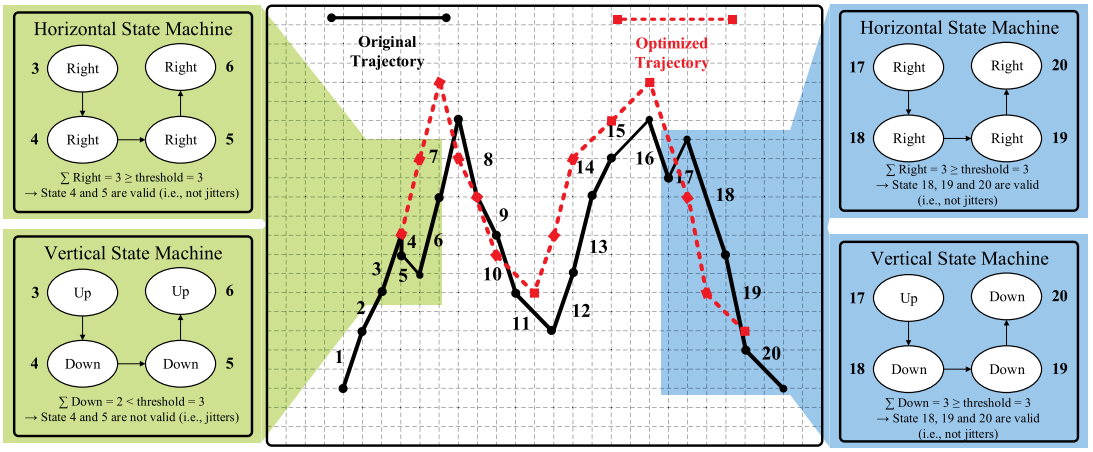


FIGURE 9. Illustration of trajectory optimization based on state machine.

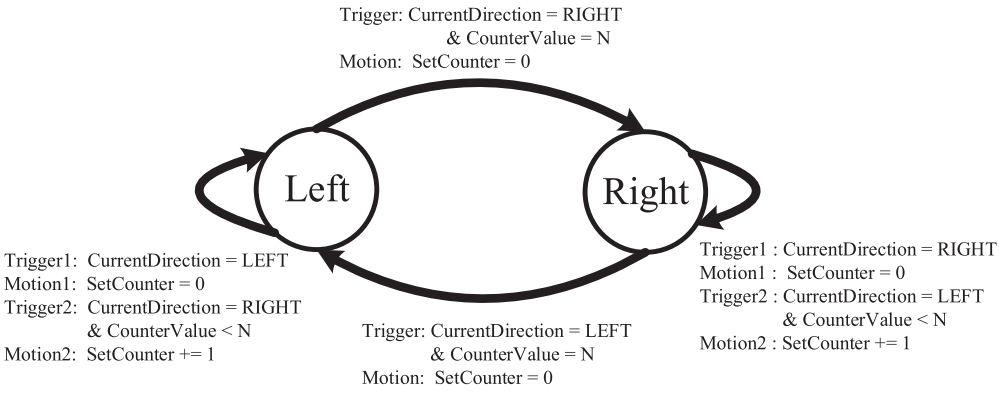


FIGURE 10. Working processes of a horizontal-direction state machine in LEFT and RIGHT direction.

3.4. Recognizing character

When the user writes a character in the air, the writing trajectory can be tracked by VPad as a stroke direction sequence. To recognize the character written in the air, VPad compares the sequence with potential stroke direction sequences of all possible characters.

To ensure the robustness of character recognition, we first divide hand movement directions into eight different categories, each of which is indexed with a number, as shown in Fig. 11a. Thus, VPad can transform a stroke direction sequence of characters into a number sequence. For example, in Fig. 11b, one stroke direction sequence of the character 'D' can be regarded as a sequence $S = [7, 3, 1, 8, 7, 5, 4]$. However, the writing stroke direction sequence varies from one user to another, so the character 'D' may be tracked as another sequence, $S = [7, 3, 8, 7, 5]$ in Fig. 11c. 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',

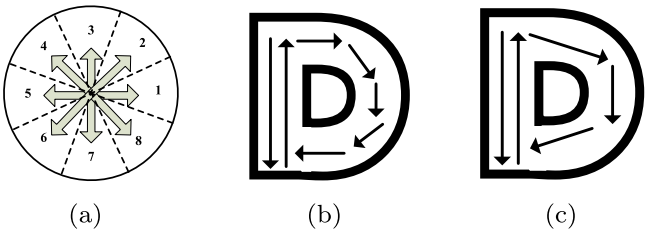


FIGURE 11. Illustration of the trajectory recognition algorithm, (a) the eight movement directions in a plane, (b) and (c) two different stroke sequences of the character 'D'.

etc, and the potential sequences of all characters compose a set, i.e. $G\{G_{char='A'}, G_{char='B'}, ...\}$.

For matching stroke direction sequences, we use the weighted minimum edit distance (WMED) [16] to measure the similarity between two sequences. The minimum edit distance between two sequences is defined as the number of operations

(*insertion*, *deletion*, *substitution*), with which one sequence is transformed into another. To improve the performance, 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

$$w = \begin{cases} |n_0 - n_1| & \text{if1} <= |n_0 - n_1| <= 4 \\ 8 - |n_0 - n_1| & \text{if5} <= |n_0 - n_1| <= 7, \end{cases} \quad (12)$$

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

For a stroke direction sequence Q extracted from a tracked trajectory, VPad scans all sequences in the list G and chooses the k-nearest-sequences of Q that have the k minimum WMED values as a set, i.e.

$$V = \{< S_1, m_1 >, < S_2, m_2 >, ..., < S_k, m_k >\}, \quad (13)$$

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 probability of stroke direction sequences being corresponding to a special character C, is

$$P_{char=c} = \frac{\sum_{S_k \in G_{char=c} \wedge <S_k, m_k> \in V} \frac{1}{m_k}}{\sum_{<S_k, m_k> \in V} \frac{1}{m_k}}. \quad (14)$$

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

4. IMPLEMENTATION

VPad tracks the vertical hand movement based on Doppler shifts of received acoustic signals, which requires to extract Doppler shifts with a high time resolution to enable continuously hand movement tracking. In this section, we present a *real-time frequency shift finding algorithm* including the *sliding-window overlap fourier transformation* and *windows function* to increase the measuring resolution for real-time tracking and the *weighted average frequency peak* to accurately estimate the Doppler shift.

4.1. Optimizing acoustic signal processing to improve the resolution

VPad transforms a time-domain signal into a frequency-domain signal using *fast Fourier transformation* (*FFT*). For 20 *kHz* and 18 *Hz* acoustic signals, VPad needs to perform FFT with the size of at least 40 000-point and 36 000-point, respectively to reach the frequency resolution of 1 *Hz*. Hence, VPad uses 40 000-point FFT in acoustic signal processing. Due to the limitation of laptops' audio device, the sampling rate is less than 44.1 *kHz*. Hence, the time interval between two FFT windows is nearly 0.9 *s* (i.e. 40 000 point/44 100 Hz). Hence, the time interval of 0.9 *s* is too long to detect the hand movement in real time.

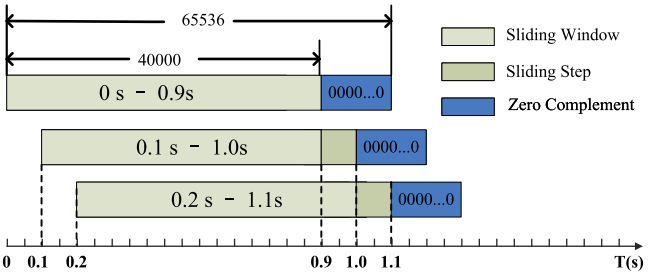


FIGURE 12. Illustration of SOFT method.

To improve the time resolution, we propose the *SOFT* method, as illustrated in Fig. 12, which uses a sliding window whose length is about 0.9 *s* with step 0.1 *s*. VPad performs FFT in each overlapped sliding window to increase the time resolution from 0.9 *s* to 0.1 *s*. Note that FFT requires the number of sampling points to be 2^n, otherwise it would result in the frequency-domain signal distortion, i.e. *fence effect* [17]. We thus add zeros at the end of each sliding-window until the number of points achieves 2^{16} to remove fence effect. After each sampling step (i.e. 0.1 *s*), VPad only keeps the sampled data collected in the latest 0.9 *s* and performs FFT. By applying SOFT method, VPad is able to track the hand movement trajectory in real time with 0.1 *s* time resolution.

4.2. Processing frequency-domain signal to relieve the frequency leak distortion

Due to frequency leakage distortion, SOFT method is difficult to achieve 1 *Hz* frequency accuracy of acoustic signal. When we derive the frequency domain of a continuous signal (i.e. an analog signal) through discrete methods, the *windowing process* is employed to process the signal, i.e. a finite-duration record of the signal is sampled from an infinite signal sequence. However, the windowing process introduces a *frequency leakage* [17] effect. This is because spurious high-frequency components are introduced into the spectrum in windowing process, which are caused by the sharp clipping of signal at the left and right ends of the window.

Due to the frequency leakage, a series of sidelobe frequency peaks appear beside the mainlobe frequency peak. This phenomenon leads to the mainlobe's energy is dispersed into the surrounding sidelobes. To deal with this problem, VPad utilizes both mainlobe and sidelobe peaks to calculate Doppler shift, even if the sidelobe peaks are caused by signal distortion rather than Doppler effect. Also, VPad adopts non-rectangular window, such as Hamming window or Caesar window [17], to suppress the influence of sidelobe peaks. After the sidelobes are eliminated by the window function, VPad can relieve the

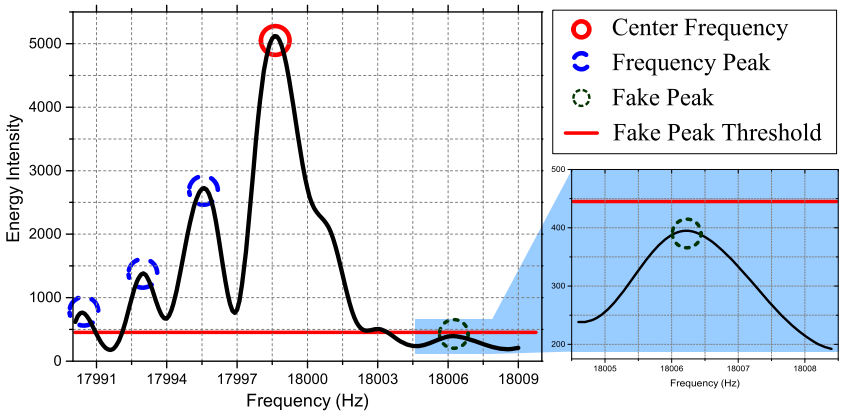


FIGURE 13. Illustration of an acoustic signal sample.

frequency leakage distortion and improve the system resolution to achieve 1 *Hz* level through *Lagrange interpolation* [17].

4.3. Analyzing frequency peak to estimate Doppler shift

In order to measure Doppler shift in each sampling window, an intuitive way is to find the *center-frequency*, i.e. the frequency point with the maximum signal intensity as shown in Fig. 13. However, since the acoustic signals are reflected through multiple paths, there exist multi-frequency peaks, as shown in Fig. 13. Hence, center frequency cannot represent the overall Doppler shift in the time interval. In order to enhance the accuracy, VPad selects all Doppler shifts caused by the hand movement to estimate the overall Doppler shift during the time interval.

Since the fastest speed of the hand movement in front of laptops is about 3.9 m/s, the targeted frequency band is about 33 *Hz* on either side of center frequency [9]. We scan the frequency signal in this field and find out all of the frequency point whose signal intensity is higher than that of neighbor points, which are *frequency peaks* as shown in Fig. 13.

However, we observe that the detected frequency peaks are not always caused by hand movement. To filter out the *fake peaks* that caused by environment noises and device fluctuates, we set a threshold on the minimum amplitude of the detected frequency peaks. VPad collects the environmental acoustic signals and the device original acoustic signals periodically. If the amplitude of detected frequency peaks is lower than either the environment noise intensity or the device noise intensity, VPad regards it as a fake peak and filters it out as shown in Fig. 13.

After removing fake peaks, VPad calculates the weighted average peak on the basis of frequency peaks. The weight of each frequency peak depends on its signal intensity, i.e.

$$f = \frac{\sum_{i=1}^{n} f_i E_i}{\sum_{i=1}^{n} E_i}, \tag{15}$$

where f_i is the frequency value of the i^{th} frequency peak and E_i is the energy intensity of the i^{th} frequency peak.

5. EVALUATION

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

5.1. 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 experimental facilities. We recruit 20 volunteers (10 males and 10 females) to conduct the experiments, whose ages are in the range of [20, 65]. In the experiments, there are 70 characters 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, $\angle$, $\wedge$, $\vee$) for users to write in the air. The experiments are conducted in three real environments, i.e. a laboratory, a noisy canteen and a moving car. In each environment, each user is required to write all characters twice with VPad on four laptops, i.e. totally 560 writings for a user. Each user writes the characters in the air with his/her own writing habit, regardless of the writing speed, the size of writing character and the writing style.

Several metrics are used in our evaluation. Assume a user actually writes a character i, while VPad recognizes it as j. Then, ρ_{ij} is 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. $Recall_k = \rho_{kk} / \sum_{i=1}^{n} \rho_{ki}$.

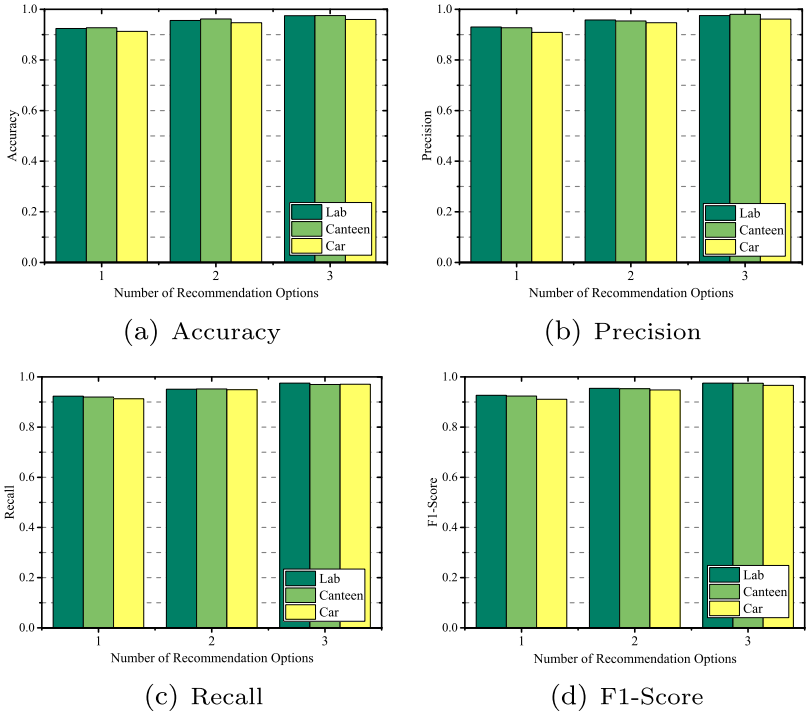


FIGURE 14. Overall performance of VPad.

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

5.2. 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.

Figure 14 shows the accuracy, precision, recall and F1-sore of VPad in three different environments. It can be observed that the performances of VPad in different environments exhibit insignificant differences. For one character recommendation option, the accuracies of VPad are all above 90% in three different environments. When VPad recommends three character options, the accuracy approaches 95% in the three environments. Meanwhile, F1-score of one, two and three character recommendation options are all above 0.9, 0.95 and 0.95, respectively in the three environments. This demonstrates VPad is robust to various environments, such as surrounding noises and vibrations.

5.3. Performance of trajectory tracking

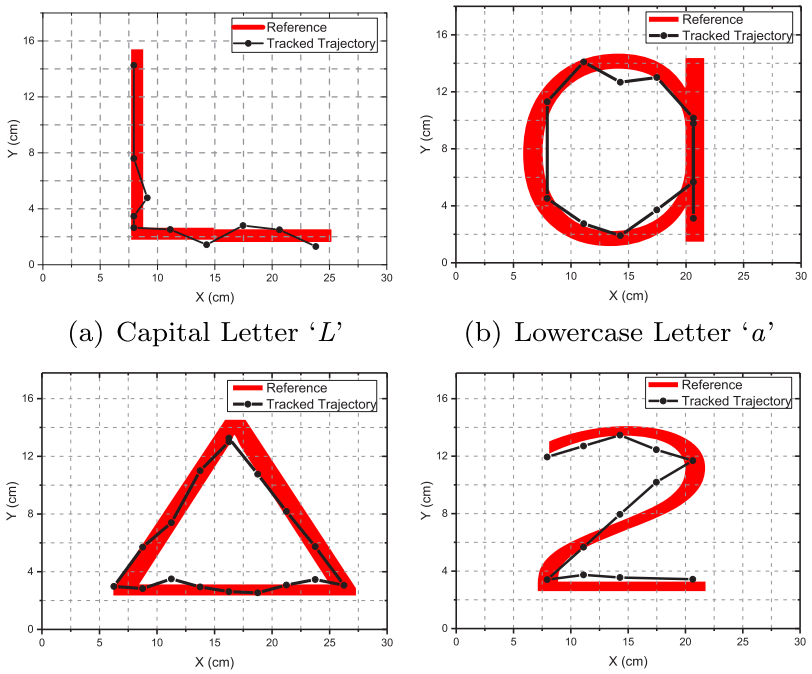
In this section, we evaluate the performance of VPad's trajectory tracking.

Writing trajectory. When a character shows on screen, a user puts the hand on the initial point of the character and then follows boundaries of the character to write it in the air. Figure 15 shows the examples of tracking trajectory and the reference trajectory when writing a character in each character set. It can be seen that the tracking trajectories are very close to the reference trajectories for all characters.

Trajectory tracking error. 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 from the point to the source-to-target connection. The average distance of all points is regarded as the error of trajectory tracking. Each user conducts the experiment 20 times. Figure 16 shows CDF of trajectory tracking error under four types of laptops. It can be seen that 80% of trajectory tracking errors are lower than 1.8 *cm*, 2.0 *cm*, 2.1 *cm* and 2.5 *cm* under 12.5-inch, 13.3-inch, 14-inch and 15.6-inch laptops, respectively. The average error of trajectory tracking is 1.55 *cm*. In addition, we conduct an experiment to evaluate the robustness of VPad against background noise, in which users write characters in the laboratory where people walk or talk around. We can see from Fig. 17 that the average error of VPad with noise is almost the same with that without noise. This is because the acoustic signal energy around the laptop is significantly higher than that elsewhere. Thus, VPad is robust to background noise.

5.4. Impact of acoustic signal processing

Two speakers emit acoustic signals every $1Hz$ among $17.5kHz \sim 18.5kHz$ and $19.5kHz \sim 20.5kHz$, respectively. Totally, 1000 acoustic signals in different frequencies are received in each



(a) Capital Letter 'L' (b) Lowercase Letter 'a'

(c) Special Character 'Δ' (d) Number '2'

FIGURE 15. Tracked trajectory of VPad and the reference trajectory.

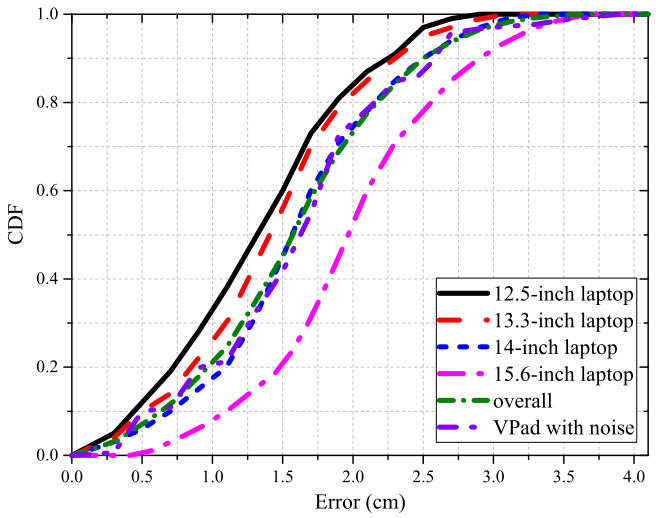


FIGURE 16. CDF of trajectory tracking error under different laptops without and with noise.

experiment. Then, VPad processes the received acoustic signals and estimates its frequency. We regard the frequency of emitted signal as the ground truth and compare the frequency error between the estimation frequency and ground truth.

Figure 17a and b show CDF of the frequency error with and without SOFT and window function. It can be seen that frequency estimations with two methods can achieve higher accuracy in both $17.5kHz \sim 18.5kHz$ and $19.5kHz \sim 20.5kHz$. Specifically, 84.5% and 93.7% of estimation errors are lower than 1 Hz in $17.5kHz \sim 18.5kHz$ and $19.5kHz \sim 20.5kHz$, respectively. And the overall average frequency error is below 0.6 Hz under both methods. Thus, SOFT and window function are effective to improve the accuracy of frequency estimation.

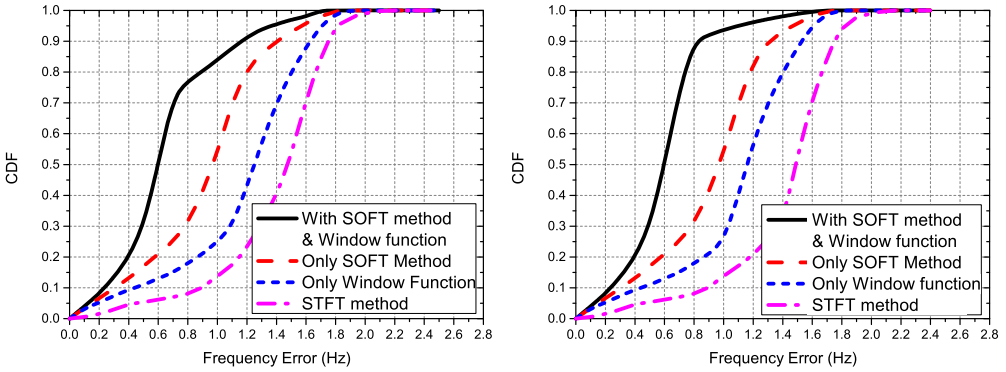
5.5. Impact of trajectory optimization

A user draws a line from an initial point to a target point in the air. To evaluate the performance of trajectory optimization, we define a ratio $D = D_a/D_s$, where D_a is the actual moving distance and D_s is the shortest path from an initial point to a specific target point. Fig. 18a shows the CDF of ratio D with and without the trajectory optimization. We can observe that 89.7% of the ratio D is lower than 1.4 with the trajectory optimization, and only 31% of the ratio D is lower than 1.4 without trajectory optimization.

Fig. 18b shows the character recognition accuracy of VPad with and without the trajectory optimization. It can be observed that the trajectory optimization algorithm improves the character recognition accuracy significantly. Specifically, the accuracy is above 93% under three recommended character options if VPad applies the optimization algorithm. Otherwise, the accuracy is only about 83%.

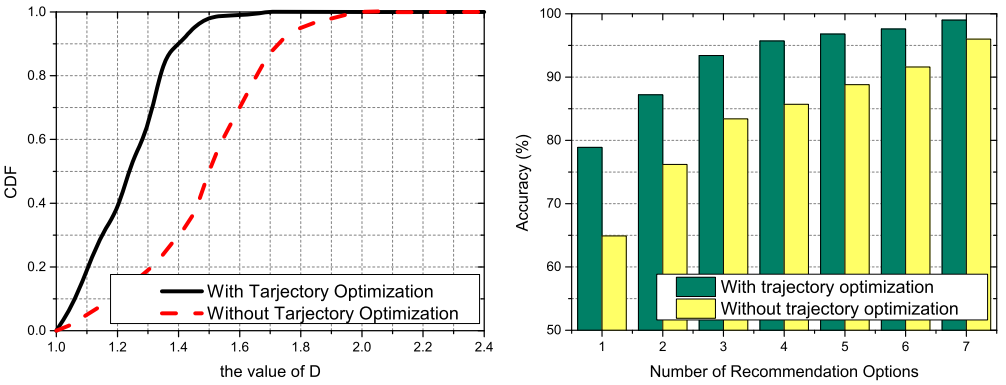
5.6. Impact of initial position error

Since VPad requires to combine the tracking approach with an initial position estimation to obtain the absolute hand position,



(a) 17.5kHz∼18.5kHz. (b) 19.5kHz∼20.5kHz.

FIGURE 17. CDF of the frequency error with and without SOFT and window function in different frequency bands.



(a) CDF of D. (b) Accuracy.

FIGURE 18. Performance of VPad with and without trajectory optimization respectively.

the error in initial position estimation affects the accuracy of VPad. Fig. 19a shows CDF of the initial position estimation error. We can see that 80% of estimation errors are lower than 1.95 *cm*, 2.0 *cm*, 2.1 *cm* and 2.6 *cm* under 12.5-inch, 13.3-inch, 14-inch and 15.6-inch laptops, respectively. The average value of estimation errors is 1.55*cm*. Also, Fig. 19b shows the accuracy of VPad under (i) the initial position is known, (ii) the initial position is known when a fixed amount of error, and (iii) our initial position estimation method. We can see that the accuracy of VPad is not sensitive to the initial position error. For example, even with 5 *cm* error, the accuracy of VPad only decreases 2.5%. Therefore, the initial position estimation of VPad is accurate for trajectory tracking.

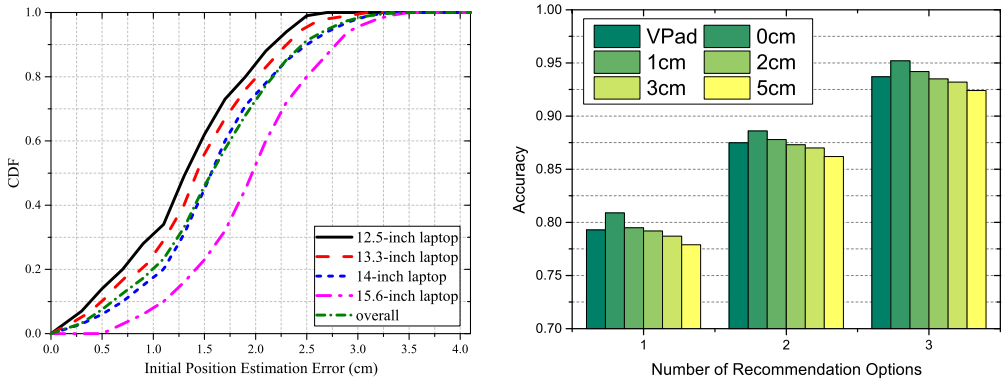
5.7. Impact of writing speed in the air

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 $|\bar{s}|$ of a character C in the air divides the writing duration Δt, i.e. $v_C = |\bar{s}|/\Delta t$. We analyze all writing speeds of 20 users and find that the distribution of the writing speed satisfies the Gaussian distribution. Thus, four 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 five 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 ($v_C \leq 5.5cm/ms$).

Figure 20 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 all below 15% under three numbers of recommendation options, respectively. This result shows that VPad is not sensitive to the writing speed of users.

5.8. Impact of writing character's size in the air

The size of writing character in the air may affect the accuracy of VPad. For each character C, we use a rectangle to surround C, and the area of the smallest rectangle $S_{char=C}$ is defined



(a) CDF of estimation error.

(b) Recognition accuracy as initial position varies.

FIGURE 19. Performance of the initial position estimation in VPad.

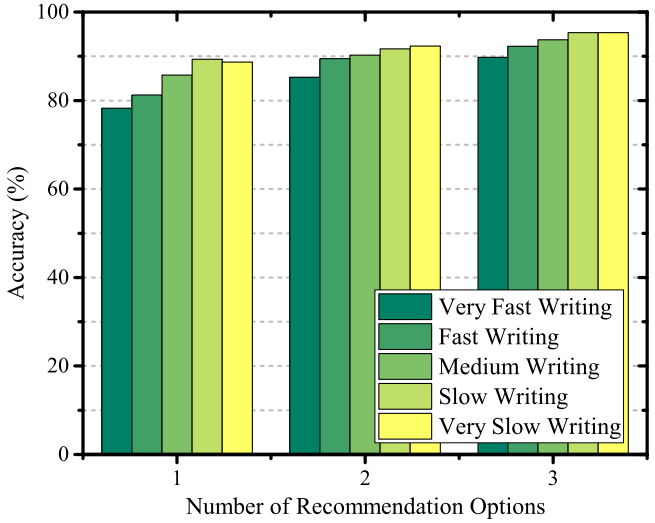


FIGURE 20. Accuracy of VPad under different writing speeds.

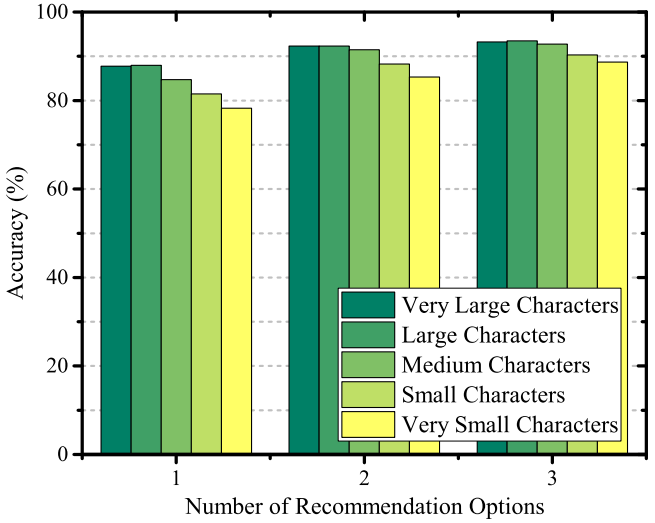


FIGURE 21. Accuracy of VPad under different character sizes.

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, four 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 five categories, i.e. very large ($P > 0.7$), large ($0.5 < P \leq 0.7$), medium ($0.3 < P \leq 0.5$), small ($0.15 < P \leq 0.3$) and very small ($P \leq 0.15$) characters.

Figure 21 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 all below 10% under three numbers of recommendation options, respectively. This result shows that VPad is insensitive to the size of writing characters.

5.9. 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 an 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

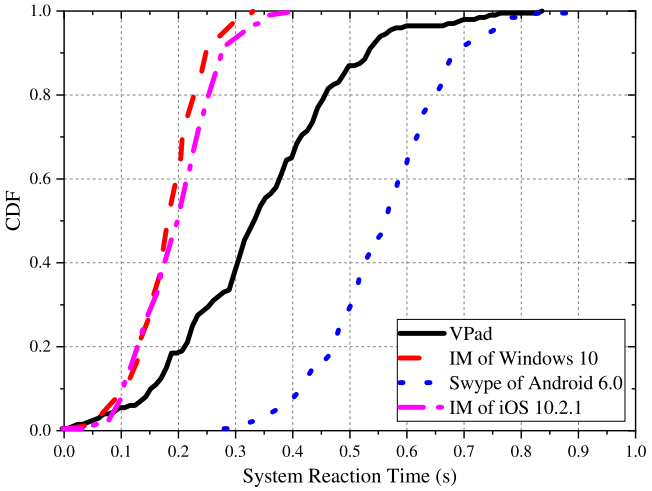


FIGURE 22. CDF of the system reaction time for VPad.

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 25 *Hz*) and the frames of the screencasts (the frame rate is 30 *Hz*), 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 $T = T_{dev} - T_e$.

Figure 22 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 0.34 *s*, 0.19 *s*, 0.56 *s* and 0.21 *s*, respectively. The system reaction time of IM in Windows 10 and iOS 10.2.1 are less because their handwriting IMs are tightly integrated in 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 0.15 *s*, which is so little for users to be aware. Also, as third-party softwares, the system reaction time of Swype is 0.7 *s* for 90% samples, but that of VPad is only 0.5 *s*. Therefore, VPad is able to achieve ideal performance and meets users' actual demand as a virtual writing tablet.

5.10. Usability study of VPad

We also ask volunteers to provide a feedback of the user experience after they finish the experiments to evaluate the usability of VPad. Specifically, we define five rating levels to measure the user experience, i.e. 1-very dislike, 2-dislike, 3-neutral, 4-like and 5-very like. After a volunteer finishes the experiments in an environment with a laptop, he/she is required to give an explicit rating that depicts the user experience during the experiments. Table 2 shows average rating of user experience

TABLE 2. Rating of user experience under different laptops in different environments.

	12.5-inch	13.3-inch	14-inch	15-inch
Laboratory	3.9	3.9	3.9	3.7
Canteen	4.1	4.0	4.0	3.8
Car	4.8	4.7	4.7	4.6

under different laptops in different environments. Overall, the ratings of user experience are all above 3.5, which indicates a good usability of VPad. We also can see that the rating in the car is significantly higher than that in other two environments. This is because users are difficult to input with keyboard in the car, due to the significant vibrations. But in other two environments, users cannot be aware of significant advance of VPad compared with keyboards. Moreover, as the size of laptop screen increases, the rating of user experience decreases. This is because as the size of laptop screen increases, a user needs to move the hand through a longer distance, which degrades the user experience.

6. DISCUSSION

In this section, we discuss the practical issues of VPad.

Controlling audio volume to avoid interference. When a user is interacting with a laptop through VPad, there may be other persons moving around in the ambient area of the laptop. Due to the omni-directional propagation of acoustic signals, these ambient persons have a certain impact on the performance of VPad. For example, VPad may mis-recognize an ambient moving person as a finger and then track the moving trajectory of the person, which leads to the wrong recognized character. Also, when two VPad are employed in a relatively short distance, similar interferences may also be introduced to the system, which reduces the performance of VPad. Fortunately, acoustic signals attenuate significantly as propagating through a distance [18, 19]. Hence, VPad can adaptively adjust the volume of transmitted signals to avoid the interference by ambient persons or adjacent VPad system while still achieving the accurate tracking and character recognition.

Generality of VPad in various laptops. VPad only utilizes the built-in audio devices in the laptops, i.e. the two speakers and one microphone, to realize the hand movement tracking and character recognition. There are several impact factors of the audio devices affecting the generality of VPad, i.e. *(i)* the number of speakers and microphones in the laptop, *(ii)* the sampling rate of the microphone and *(iii)* the layout of the speaker and microphone in the laptop. First, most laptops have stereo speakers and at least one microphone [20]. Advanced laptops even deploy more speakers and microphones. Hence, there are enough audio devices for VPad deployment. Second, since VPad emits 18 *kHz* and 20 *kHz* acoustic signals for hand movement tracking, the microphone should have a sampling

rate over 40 *kHz*, according to Nyquist theorem. Fortunately, the built-in microphone of laptops is usually capable with a sampling rate of 44.1 *kHz*, which is larger than 40 *kHz*. Hence, the sampling rate of microphone in most laptops satisfies the requirement of VPad. Finally, the layout of the audio devices varies among different laptops. However, the layout of audio devices in the laptop can be pre-determined with the specific laptop model. Hence, VPad can be generalized to different laptops with the given layout of audio devices.

Noise cancellation with dual microphone. In this work, we only use one microphone of the laptop to pick up ultrasound signals for tracking hand movements. Recent commercial laptops are equipped with multiple microphones, which are used for stereo recording and noise reduction. Due to different deployment positions of the microphones on the device, the signals recorded by the microphones came from different propagation paths and contain different properties of channel fading effects, background noise levels, etc. It thus has great potential to use multiple microphones to perform noise reduction and recording-quality calibration, so as to improve tracking performance. We leave this in our future work.

Extension of VPad to smartphones. Recently, the smartphones are capable with more powerful sensors. Existing studies employ inbuilt sensors to infer user interest during browsing [21], sense driving condition [22] and improve energy efficiency for mobile devices [23]. The inbuilt audio devices are also gradually enhanced. For example, Samsung Galaxy S9 is equipped with stereo speakers [24] to support better digital audio video experience, which provides the opportunity to extend VPad to smartphones. Although smartphones are capable with touch screens, VPad further enables the smartphone with the in-air interaction capability, which supports the gradually-prevalent AR/VR applications on smartphones. This exhibits the tremendous vitality of VPad in supporting various mobile devices.

7. RELATED WORK

Existing studies on hand movement tracking and handwriting recognition are categorized as follows.

Vision-based tracking: commercial tracking systems, like LeapMotion [5], Kinect [6] and Wii [25], use depth and infrared cameras to extend the user–device interactions. However, they can only be implemented in a particular optical manner (i.e. LOS) that is easy to be interfered, and built-in cameras of laptops are usually with low resolution, which cannot precisely track hand movements.

RF-based tracking: RF-based sensing techniques have been driven by the emerging trend of RF-based (e.g. RFID, WiFi) devices. For example, ArrayTrack [26] and RF-IDraw [7] employed the phase of received RFID signals to track the user's movement trajectory. WiSee [27] and WiDraw [8] utilized channel state information (CSI) and minute Doppler shifts extracted from WiFi signals to achieve fine-grained gesture recognition, respectively. However, these approaches either need an additional device, such as RFID readers/tags or are easily affected by the environmental interferences such as people walking by.

Motion sensor-based tracking: some recent works use inertial sensors (e.g. accelerometer and gyroscope) embedded in mobile devices or wearables to capture human motions. For instance, [28] could track the phone's movement and recognize the English character written by users. Shen *et al.* [29] and Wang *et al.* [30] demonstrated that smart watches can track the user's arm motions even know what the user is typing on a keyboard. However, all these systems require an external device such as a smartphone or a wearable.

Acoustic signal-based tracking: some existing researches rely on acoustic signals, such as keystroke snooping [18, 31, 32], human's sleep apnea situation monitoring [33], user authentication [34, 35] and indoor localization [36]. Regarding the motion tracking, SoundWave [9], Airlink [37] and SurfaceLink [38] employed Doppler effect of acoustic signals and surface-mounted piezoelectric sensors to recognize gestures, which can only provide well-defined gesture recognition instead of the accurate position tracking. AAmouse [10] realized a virtual mouse in the air using acoustic signals and CAT [11] developed a high-precision tracker based on acoustic signals. But both of them can only track hand movement through an additional acoustic-signal emitter from a smartphone that is held in the user's hand. Additionally, LLAP [12] and FingerIO [13] designed trajectory tracking algorithms for wearable devices to track users' fingers near the wearable devices, and Strata [14] developed a fine-grained acoustic-based tracker for smartphones using the channel impulse response of acoustic signals. Since the audio devices in laptops are different from that in wearable devices and smartphones, all these works cannot be deployed in laptops. Moreover, LLAP and FingerIO employ CW signals and OFDM pulses to achieve better tracking accuracy, respectively, which are susceptible to the interference of background movements, so that both of them cannot works well for tracking hand movements in a larger distance.

Handwriting recognition: handwriting recognition has been widely studied in past decades. Except for handwriting recognition on specific devices [39, 40], most recent studies focus on offline handwriting recognition, i.e. recognizing a character after users' writing [41, 42]. However, these works can only recognize characters in handwriting images. Since the tracked hand movements are stroke direction sequences instead of images, these approaches cannot be deployed to recognize characters after users write in the air.

8. CONCLUSIONS

In this paper, we design a virtual writing tablet for laptops, VPad, which can accurately track hand movements and recognize characters written in the air leveraging acoustic signals.

Unlike existing work, VPad only utilizes two speakers and one microphone, which are built-in on most commercial laptops, to track users' hand movement trajectories in the air without intrusive wearable sensors or additional infrastructures and employs a light-weight method to recognize the handwriting characters in real time. Firstly, we have explained how a user's hand movements can be tracked without intrusive wearable sensors by instead detecting the peaks of the acoustic signal and its Doppler shifts. Secondly, we achieve higher resolution Doppler shift measurement in real time by developing a sliding-window overlapping Fourier transform. Finally, a stroke direction sequence model based on probability estimation is employed to achieve accurate character recognition. Extensive experiments demonstrate the feasibility and effectiveness of VPad. Moving forward, we are interested in extending our implementation to more device-free human-computer interaction, such as virtual reality and augmented reality, etc.

Funding

National Nature Science Foundation of China [61772338].

Acknowledgments

We would like to sincerely thank the anonymous reviewers and editors for their helpful suggestions and comments to improve the quality of this paper.

REFERENCES

[1] Research, P.M. (2018) Touchscreens in mobiledevices market:global industry analysis and forecast. http://www.persistencemarketresearch.com/ (last accessed at 1 October 2018).
[2] Drawboard (2018) Drawboard pdf. https://www.drawboard.com/pdf/ (last accessed at 1 October 2018).
[3] Lenovo. (2018) Writeit turns your touchscreen into a canvas. http://www.lenovo.com/us/en/apps/writeit/ (last accessed at 1 October 2018).
[4] Lowell, U. (2011) Multi-touch screen helps the disabled. https://www.uml.edu/News/stories/2010-11/student touch screen.aspx (last accessed at 1 October 2018).
[5] Codd-Downey, R. and Stuerzlinger, W. (2014) Leaplook: a free-hand gestural travel technique using the leap motion finger tracker. In *Proc. of ACM SUI*, Honolulu, HI, USA, October 153. ACM-New York, NY.
[6] Han, J., Shao, L., Xu, D. and Shotton, J. (2013) Enhanced computer vision with microsoft kinect sensor: a review. *IEEE T. Cybernetics*, 43, 1318–1334.
[7] Wang, J., Vasisht, D. and Katabi, D. (2014) Rf-idraw: virtual touch screen in the air using rf signals. In *Proc. of ACM SIGCOMM*, Chicago, IL, USA, August, pp. 235–246. ACM-New York, NY.
[8] Sun, L., Sen, S., Koutsonikolas, D. and Kim, K.H. (2015) Widraw: enabling hands-free drawing in the air on commodity wifi devices. *Proc. of ACM MobiCom*, Paris, France, May, pp. 77–89. ACM-New York, NY.
[9] Gupta, S., Morris, D., Patel, S. and Tan, D. (2012) *Soundwave: using the doppler effect to sense gestures. Proc. of ACM CHI, Austin, Texas, May*, pp. 1911–1914. ACM-New York, NY.
[10] Yun, S., Chen, Y.C. and Qiu, L. (2015) Turning a mobile device into a mouse in the air. *Proc. of ACM MobiSys*, Florence, Italy, May, pp. 15–29. ACM-New York, NY.
[11] Mao, W., He, J. and Qiu, L. (2016) Cat: high-precision acoustic motion tracking. *Proc. of ACM MobiCom*, New York, NY, October, pp. 69–81. ACM-New York, NY.
[12] Wang, W., Liu, A.X. and Sun, K. (2016) Device-free gesture tracking using acoustic signals. *Proc. of ACM MobiCom*, New York, NY, October, pp. 82–94. ACM-New York, NY.
[13] Nandakumar, R., Iyer, V., Tan, D. and Gollakota, S. (2016) Fingerio: using active sonar for fine-grained finger tracking. *Proc. of ACM CHI*, San Jose, CA, USA, May, pp. 1515–1525. ACM-New York, NY.
[14] Yun, S., Chen, Y.-C., Zheng, H., Qiu, L. and Mao, W. (2017) Strata: fine-grained acoustic-based device-free tracking. *Proc. of ACM MobiSys*, Niagara Falls, NY, USA, June, pp. 15–28. ACM-New York, NY.
[15] Kannan, P.G., Venkatagiri, S.P., Chan, M.C., Ananda, A.L. and Peh, L.S. (2012) Low cost crowd counting using audio tones. *Proc. of ACM Sensys*, Toronto, ON, Canada, November, pp. 155–168. ACM-New York, NY.
[16] Deshpande, P.S., Malik, L. and Arora, S. (2008) Fine classification and recognition of hand written devnagari characters with regular expressions and minimum edit distance method. *J. Comput.*, 3, 11–17.
[17] Oppenheim, A.V. and Schafer, R.W. (1999) Discrete-Time Signal Processing. *Prentice Hall Signal Processing*, 23, 1–39.
[18] Lu, L., Yu, J., Chen, Y., Zhu, Y., Xu, X., Xue, G. and Li, M. (2019) Keylistener: inferring keystrokes on qwerty keyboard of touch screen through acoustic signals. *Proc. of IEEE INFOCOM*, Paris, France, April, pp. 775–783. IEEE-Piscataway, NJ.
[19] Yu, J., Lu, L., Chen, Y., Zhu, Y. and Kong, L. (2019) *An indirect eavesdropping attack of keystrokes on touch screen through acoustic sensing*, pp. 1–14. IEEE Trans. Mob. Comput., Early Assess. 19, 1–15
[20] Geek.com (2019). Laptop speakers. https://www.geek.com/laptop-speakers/ (last accessed 8 May 2019).
[21] Lu, L., Yu, J., Chen, Y., Zhu, Y., Li, M. and Xu, X. (2019) I3: sensing scrolling human-computer interactions for intelligent interest inference on smartphones. *Proc. of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3, 97: 1–97:22.
[22] Wu, Z., Li, J., Yu, J., Zhu, Y., Xue, G. and Li, M. (2016) L3: sensing driving conditions for vehicle lane-level localization on highways. *Proc. of IEEE INFOCOM*, San Francisco, CA, USA, April, pp. 1–9. IEEE-Piscataway, NJ.
[23] Yu, J., Han, H., Zhu, H., Chen, Y., Yang, J., Zhu, Y., Xue, G. and Li, M. (2015) Sensing human-screen interaction for energy-efficient frame rate adaptation on smartphones. *IEEE Trans. Mob. Comput.*, 14, 1698–1711.
[24] TechRadar (2019) Samsung galaxy s9 plus review. https://www.techradar.com/reviews/samsung-galaxy-s9-plus-review/3 (last accessed at 8 May 2019).

[25] Nintendo (2018) Nintendo wii. http://www.nintendo.com/wii (last accessed at 1 October 2018).
[26] Xiong, J. and Jamieson, K. (2013) Arraytrack: a fine-grained indoor location system. *Proc. of USENIX NSDI*, Boston, MA, USA, April, pp. 71–84. ACM-New York, NY.
[27] Pu, Q., Gupta, S., Gollakota, S. and Patel, S. (2013) Whole-home gesture recognition using wireless signals. *Proc. of ACM MobiCom*, Miami, FL, USA, September, pp. 27–38. ACM-New York, NY.
[28] Agrawal, S., Constandache, I., Gaonkar, S., Roy Choudhury, R., Caves, K. and Deruyter, F. (2011) Using mobile phones to write in air. *Proc. of ACM MobiSys*, Bethesda, MD, USA, June, pp. 15–28. ACM-New York, NY.
[29] Shen, S., Wang, H. and Roy Choudhury, R. (2016) I am a smartwatch and i can track my user's arm. *Proc. of ACM MobiSys*, Singapore, June, pp. 85–96. ACM-New York, NY.
[30] Wang, H., Lai, T.T. and Choudhury, R.R. (2015) Mole: motion leaks through smartwatch sensors. *Proc. of ACM MobiCom*, Paris, France, September, pp. 155–166. ACM-New York, NY.
[31] Liu, J., Wang, Y., Kar, G., Chen, Y., Yang, J. and Gruteser, M. (2015) Snooping keystrokes with mm-level audio ranging on a single phone. *Proc. of ACM MobiCom*, Paris, France, September, pp. 142–154. ACM-New York, NY.
[32] Wang, J. *et al.* (2014) Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization. *Proc. of ACM MobiSys*, Bretton Woods, NH, June, pp. 14–27. ACM-New York, NY.
[33] Nandakumar, R. *et al.* (2015) Contactless sleep apnea detection on smartphones. *Proc. of ACM MobiSys*, Florence, Italy, May, pp. 45–57. ACM-New York, NY.
[34] Lu, L., Yu, J., Chen, Y., Liu, H., Zhu, Y., Liu, Y. and Li, M. (2018) Lippass: lip reading-based user authentication on smartphones leveraging acoustic signals. *Proc. of IEEE INFOCOM*, Honolulu, HI, USA, April, pp. 1466–1474. IEEE-Piscataway, NJ.
[35] Lu, L., Yu, J., Chen, Y., Liu, H., Zhu, Y., Kong, L. and Li, M. (2019) Lip reading-based user authentication through acoustic sensing on smartphones. *IEEE/ACM Trans. Netw.*, 27, 447–460.
[36] Huang, W. *et al*. (2014) Shake and walk: acoustic direction finding and fine-grained indoor localization using smartphones. *Proc. of IEEE INFOCOM*, Toronto, Canada, April, pp. 370–378. IEEE-Piscataway, NJ.
[37] Chen, K. Y., Ashbrook, D., Goel, M., Lee, S. H. and Patel, S. (2014) Airlink: sharing files between multiple devices using in-air gestures. *Proc. of ACM UbiComp*, Seattle, USA, September, pp. 565–569. ACM-New York, NY.
[38] Goel, M., Lee, B., Islam Aumi, M. T., Patel, S., Borriello, G., Hibino, S. and Begole, B. (2014) Surfacelink: using inertial and acoustic sensing to enable multi-device interaction on a surface. *Proc. of ACM CHI*, Toronto, Canada, April, pp. 1387–1396. ACM-New York, NY.
[39] Nouboud, F. and Plamondon, R. (1990) On-line recognition of handprinted characters: survey and beta tests. *Pattern Recognit.*, 23, 1031–1044.
[40] Tappert, C. C., Suen, C. Y. and Wakahara, T. (1988) Online handwriting recognition-a survey. *Proc. of IEEE ICPR*, Rome, Italy, November, pp. 1123–1132. IEEE-Piscataway, NJ.
[41] Bahlmann, C., Haasdonk, B. and Burkhardt, H. (2002) Online handwriting recognition with support vector machines—a kernel approach. *Proc. of IWFHR*, Niagara-on-the-Lake, Ontario, Canada, August, pp. 49–54. IEEE-Piscataway, NJ.
[42] Graves, A. and Schmidhuber, J. (2009) Offline handwriting recognition with multidimensional recurrent neural networks. *Adv. Neural Inf. Process. Syst.*, 21, 545–552.