

QwertyRing: Text Entry on Physical Surfaces Using a Ring

YIZHENG GU, CHUN YU*, ZHIPENG LI, ZHAOHENG LI, XIAOYING WEI, and YUANCHUN SHI, Tsinghua University, China

The software keyboard is widely used on digital devices such as smartphones, computers, and tablets. The software keyboard operates via touch, which is efficient, convenient, and familiar to users. However, some emerging technology devices such as AR/VR headsets and smart TVs do not support touch-based text entry. In this paper, we present QwertyRing, a technique that supports text entry on physical surfaces using an IMU (Inertial Measurement Unit) ring. Users wear the ring on the middle phalanx of the index finger and type on any desk-like surface, as if there is a QWERTY keyboard on the surface. While typing, users do not focus on monitoring the hand motions. They receive text feedback on a separate screen, e.g., an AR/VR headset or a digital device display, such as a computer monitor. The basic idea of QwertyRing is to detect touch events and predict users' desired words by the orientation of the IMU ring. We evaluate the performance of QwertyRing through a five-day user study. Participants achieved a speed of 13.74 WPM in the first 40 minutes and reached 20.59 WPM at the end. The speed outperforms other ring-based techniques [24, 30, 45, 68] and is 86.48% of the speed of typing on a smartphone with an index finger. The results show that QwertyRing enables efficient touch-based text entry on physical surfaces.

CCS Concepts: • Human-centered computing → Text input; Gestural input.

Additional Key Words and Phrases: smart ring, text entry, touch input

ACM Reference Format:

Yizheng Gu, Chun Yu, Zhipeng Li, Zhaoheng Li, Xiaoying Wei, and Yuanchun Shi. 2020. QwertyRing: Text Entry on Physical Surfaces Using a Ring. *Proc. ACM Meas. Anal. Comput. Syst.* 37, 4, Article 111 (August 2020), 29 pages. <https://doi.org/10.1145/1122445.1122456>

1 INTRODUCTION

Text entry for head-mounted displays (HMDs), such as AR and VR headsets requires improvement. Existing approaches for text entry leverage head rotation [73], mid-air gesture [24], or swiping on the headset [20, 74] for inputting text. However, text entry techniques based on head rotation and mid-air gesture create fatigue, while swiping on the headset to input text is not efficient enough (<10 WPM). These solutions are not user friendly and efficient compared to the widely used software keyboard on smartphones, which is based on "touch" - the most frequently used input method. In our research, we attempted to support touch-based text entry on any physical surface using a single wearable device.

We present QwertyRing, a text entry technique using a finger-worn 6-axis IMU. QwertyRing enables text entry on any desk-like surface, which is rigid, flat and spacious enough to accommodate a keyboard. Users enter

*This is the corresponding author

Authors' address: Yizheng Gu, guyz17@mails.tsinghua.edu.cn; Chun Yu, chunyu@tsinghua.edu.cn; Zhipeng Li, zp-li16@mails.tsinghua.edu.cn; Zhaoheng Li, zhaohengli.thu@gmail.com; Xiaoying Wei, wei-xy17@mails.tsinghua.edu.cn; Yuanchun Shi, shiyc@tsinghua.edu.cn, Key Laboratory of Pervasive Computing, Ministry of Education, Department of Computer Science and Technology, Tsinghua University, Beijing, China.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2020 Association for Computing Machinery.

2476-1249/2020/8-ART111 \$15.00

<https://doi.org/10.1145/1122445.1122456>

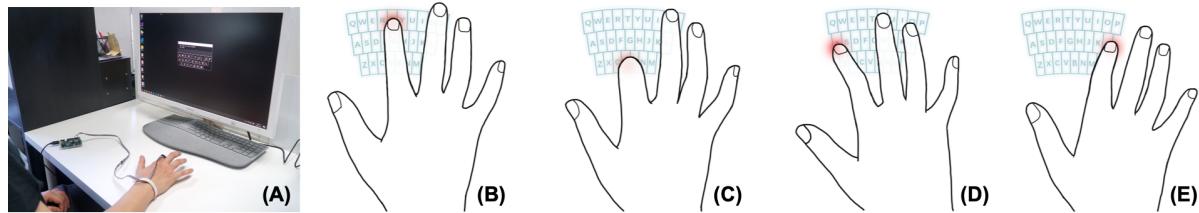


Fig. 1. A user wears the ring on the middle phalanx of the index finger and rests his wrist on a physical surface (Figure A). He imagines that there is a QWERTY keyboard in the reachable area of his index finger (Figure B-E), and then types. QwertyRing predicts the user’s desired words according to the motion data. The user receives visual feedback from a digital device display, such as a computer monitor.

characters and select the desired words from word suggestions. To use QwertyRing, users wear the ring on the middle phalanx of the index finger and receive visual feedback from an external display, e.g., an AR/VR headset or a smart TV. Users rest the wrist on the interaction surface and imagine that there is a QWERTY keyboard in the reachable area of the index finger, and then type. The system detects touch events on the surface and predicts users’ desired words according to the orientation of the IMU at touch moments.

QwertyRing has multiple inherent advantages. First, QwertyRing gathers data exclusively from an IMU sensor, which is affordable to produce and low power consumption. Second, QwertyRing is convenient to use. The device is wearable and can be used on any desk-like surface. Third, typing with QwertyRing requires no visual attention to the hand. Users can focus on the display, which is important in VR and smart TV scenarios [35]. Finally, QwertyRing is easy to learn: it enables direct touch input on the QWERTY layout, which retains users’ muscle memory based on daily use of smartphone typing.

In literature, the feasibility of typing on physical surfaces using a ring is unexplored. There are some difficulties. First, 2D finger tracking using only a ring is often not accurate (error > 1 cm), even with personalized calibration [29]. Second, while accurate detection of touch down on physical surfaces using a ring was proposed recently [21], the detection of more touch events including touch up, long press and swiping were unexplored. To address these gaps and build QwertyRing, we conducted three user studies, each contributing to answer one of the three research questions as follows:

- (1) **RQ1:** *how to detect touch events on physical surfaces using a ring?* We conducted the first study to collect users’ data of touch interactions including tapping, long presses and swiping. Using this data, we followed prior work [21] to detect touch down on physical surfaces and proposed our method to detect touch up event. The accuracies (F_1 score) were 99.30% and 99.50% respectively. Based on the sensing of touch down and touch up, the system can accurately recognize other touch events, including long press, swiping left and swiping right by threshold methods.
- (2) **RQ2:** *what is an optimal design for the text entry decoder?* We conducted the second study to collect users’ typing data. Based on the data, we analyzed the user behavior of typing and designed the decoder of QwertyRing. Combined with a trigram [28] language model, the decoder reached a top-1 accuracy of 79.0% and a top-5 accuracy of 94.6% in simulation. The result shows that the decoder could support text entry with word suggestions. Besides, there was a large difference in user behavior, which inspired us to explore the personalization.
- (3) **RQ3:** *how fast can a user type with QwertyRing?* We conducted the third study to evaluate the performance of QwertyRing with the general model and the personal model through a five-day long, between-subject study. Both the two models (decoders), were fitted by all of the data in the second study. The personal model updated every day using existing personal data in the third study, while the general model did not change

over days. The result shows that both of the two models performed well. In average, participants typed at a speed of 13.74 WPM in the first 40 minutes and reached 20.59 WPM with a five-day training. There was a trend that the personal model decoded better than the general model from day three on ($p = 0.09$).

The contributions of this work are three-fold:

- (1) We support touch-based and QWERTY-based text entry on any physical surface using a single wearable device.
- (2) We follow prior work [21] to sense touch down on physical surfaces using an IMU ring and supplement it with the touch up sensing. The accuracies are higher than 99%. We further detect touch events including long press, swiping left and swiping right.
- (3) We empirically demonstrate that QwertyRing is efficient, accurate and easy to use.

2 RELATED WORK

2.1 Wearable Text Entry

There has been a growing interest in wearable text entry, which is useful in multiple use case scenarios such as AR/VR headsets use [73, 74], large displays use [39, 57], smartwatch use [48, 72] and so on.

2.1.1 Smartwatch Text Entry. Embedding a touch screen on the body is a straightforward way to develop wearable text entry. The smartwatch is a widely used platform. The "fat finger problem" is a challenge for smartwatch text entry [71]: given the limited screen space, the individual keys are too small to be touched accurately. For character-level text entry, some techniques use iterative selection [11, 27, 48] or finger identification [23] to overcome the "fat finger problem", while others adopt strategies to avoid finger occlusion [34, 59]. Some text entry techniques offer word suggestions to achieve a faster performance [9, 18, 65]. However, there is a tradeoff between screen size and speed of inputting text [71]. Moreover, smartwatch text entry based on touch screens, face the problems of bimanual interaction and affordability.

Some smartwatch text entry techniques do not leverage the touch screen. COMPASS [72] enables text entry by rotating the bezel of a smartwatch and reports a speed of 12.5 WPM. WrisText [16] enables one-handed text entry on smartwatches using wrist gestures and reports a speed of 15.2 WPM. These techniques are not easy to learn and apply because they do not follow widely used configurations (such as QWERTY keyboards and touch interactions).

2.1.2 Mid-air Text Entry. Most mid-air text entry techniques track hand movement by optical methods. They are applicable in wearable scenarios, e.g., using front cameras of AR headset for hand tracking. Hand tracking supports mid-air text entry based on graffiti [43], selection [39], WGK (Word Gesture Keyboard) [31] and handwriting [57]. QWERTY is the best layout among mid-air selection based text entry, for generating the quickest WPM (18.9 WPM) [39]. WGK is suitable in mid-air inputting (28.1 WPM) [40]. However, these techniques might lead to fatigue. Furthermore, prior work needed expensive devices (e.g., Optitrack [52]) for accurate hand tracking. Thus, it is not practical to support mid-air text entry in commercial AR headsets.

2.1.3 Glove-based Text Entry. Many researchers design digital gloves specifically for text entry. There are two types of techniques for glove-based text entry, which leverage (1) mid-air handwriting and (2) tapping between fingers to input text. (1) Techniques based on mid-air handwriting are easy-to-learn [1, 80]. However, they have problems of fatigue and the upper limitation of 15 WPM in handwriting [13]. (2) Other techniques utilize tapping between fingers to input characters [6, 15, 51, 55, 58, 66]. Most of them rely on specific inputting rules, so the learning effect is a big problem. DigiTouch [66] is a clever approach that maps a split-QWERTY keyboard layout to the users' fingers and uses thumb-to-finger touch in order to input text. DigiTouch users learned to type reliably and achieved a mean typing speed of 16.0 WPM after three hours of practice.

2.1.4 Ring-based Text Entry. Few studies were conducted to explore the effectiveness of ring-based text entry. Some of the studies use finger-worn sensors to recognize hand gestures for inputting the 26 characters [22, 75, 78]. However, these works focus on the accuracy of hand gesture recognition (none of which are more than 93% accurate) but did not report text entry speed. ThumbText [30] deploys a small touch screen on the index finger. Users tap on the touch screen using the thumb in two steps to input a character. The expert speed is 11.4 WPM. TipText [68] features a miniature QWERTY keyboard residing invisibly on the first phalanx of the index finger. The keyboard layout was a 2 by 3 grid optimized for eyes-free input. The expert speed is 13.3 WPM. RotoSwype is a mid-air word-gesture typing based on the orientation of a finger-worn IMU [24]. Users press a button to start drawing and release to start candidate selection. The speed is 14.8 WPM after a five-day training.

The TypingRing [45] most closely resembles our research. With TypingRing, users type on a surface in a two-step strategy. TypingRing divides the QWERTY keyboard into nine zones, each of which contains three keys. Users first select a zone by moving their hand and then identifying the correct key by tapping with the index, middle or ring finger. The expert speed is 10.0 WPM. However, TypingRing has a number of major limitations. First, the two-step inputting strategy is not straightforward. Second, there is no experiment details in the paper. Third, the accuracy of touch detection was only 80%, which is not usable on a larger scale.

2.1.5 Summary. In addition to the studies mentioned above, there are other input modalities that can be utilized in text entry through wearable devices. For example, head rotation [73], tilting [50], pressure [79], gaze [32, 38, 56] and 1D touch screen [74] can be used for text entry. However, touch-based and QWERTY-based keyboards are still the most commonly used text entry system on current smart devices. We attempt to enable this experience on a physical surface using only a wearable device.

2.2 Ring Interaction

Smart rings are a rapidly growing field in the HCI domain. Digital Digits [60] conducts a comprehensive survey of finger augmented devices and points out three functions of smart rings. A primary function of smart rings is to sense beyond what the ordinary human finger senses (e.g., gesture recognition [25, 46, 63, 64, 76, 78]). Second, smart rings provide information to the wearer (e.g., tactile feedback [53, 61]). Thirdly, smart rings control or send information via fingers to an external object (e.g., manipulation [3, 8, 26, 42]). Most studies (including QwertyRing) leverage the ring as a sensing device. In this section, we discuss the sensing ability of smart rings in two aspects: finger tracking and touch detection.

2.2.1 Finger Tracking. Acoustic sensing [77] and magnetic field (MF) sensing [8, 10, 49] techniques track the relative position of finger ring to an additional hand-worn device (e.g., microphones in the smartwatch or a magnet in another ring). The accuracies are 1.3 cm, 2.5 mm, 4.8 mm, and 4.4 mm respectively. AuraRing is the latest wearable magnetic tracking system designed for tracking finger movement [49]. It has a resolution of 0.1 mm and a dynamic accuracy of 4.4 mm. However, AuraRing consists of not only a ring but also a wristband with multiple sensor coils, which make it not as practical as a ring-only system. Optical techniques use cameras on the hand to track the fingers [7, 62, 69]. They are problematic due to occlusion, high power needs and computing resource consumption. IMU can be integrated into smart rings. Smart rings are low-cost, small and have low power consumption. MIDS [33] combine a smartwatch with two IMU rings to sense the location of a fingertip on physical surfaces. LightRing [29] tracks the fingertip on surfaces. The device consists of an infrared proximity sensor for measuring finger flexion and a gyroscope for measuring finger rotation. However, these works involve more than a single sensor, have low accuracy and need calibration. In our work, we tried to estimate the flexion and rotation of fingers using a single 6-axis IMU.

2.2.2 Touch Detection. It is straightforward to detect touch on physical surfaces by finger-worn IMU sensors [33]. Most related works did not focus on touch detection or achieve a satisfying accuracy. They focused on

finger tracking (relative motion) [47], touch finger identification [41] or touch surface identification [62] instead. These work used simple threshold methods to sense touch events [33, 44, 47], yielding an accuracy of up to 89.8%. Recently, researchers proposed a machine learning method to sense touch down on any physical surface with finger-worn IMU sensor, and reached an accuracy of 98.6% [21]. However, this method can only detect touch down. The method can not accurately detect touch up, which is an important supplement to touch down that enables many basic touch gestures such as long-press and swipe. Like many other wearable text entry techniques [68, 71, 74], QwertyRing needs multiple touch gestures including long-press and swipe to completely support the text entry application. So in order to build QwertyRing, we need to solve the problem of touch up detection first.

2.2.3 Summary. (1) finger tracking using a single ring is not accurate (error > 1 cm) even with personalized calibration [29]; (2) Accurate touch down detection was achieved in 2019 [21], while touch up detection was unexplored. Given these shortcomings, the feasibility of QwertyRing (typing with an IMU ring) was unknown. We had to solve two problems to build QwertyRing: (1) To predict users' desired words from inaccurate input using Bayesian decoder. Like many other text entry techniques with inaccurate input [16, 18, 24, 40], QwertyRing could not support Out of Vocabulary (OOV) words. (2) To detect touch events on any physical surface, including touch up, long press and swiping.

2.3 Bayesian Decoder

The Bayesian approach [17] is a widely used text entry decoder. This approach allows users to input words without aiming carefully. The basic idea is to estimate the probability of each candidate word based on the word frequency (language model) and the touchpoint distribution (touch model). For the language model, our work evaluates unigram, bigram, and trigram models [28], which predict a word from previous words in an input sequence. For the touch model, most text entry techniques assumed that touchpoints on a single key obey a Gaussian distribution (absolute model) [4, 17]. QwertyRing adopts the absolute model on the mapping from finger flexion to the Y-axis of the keyboard, because users flex the index finger to input keys on different rows of the keyboard. BlindType [35] revealed that the vectors between successive touchpoints also obey a Gaussian distribution (relative model) when users receive text feedback on a separate screen. QwertyRing adopts the relative model on the X-axis of the keyboard. It leverages the angle difference between successive tapping, which is suitable for the mental model of eyes-free typing.

3 QWERTYRING

QwertyRing was designed with an iterative process. In this section, we first introduce the setting and usage of QwertyRing. Then, we share motivations for the design of QwertyRing. The device of QwertyRing is a 6-axis IMU worn on the middle phalanx of the index finger. As figure 1A shows, the user rests the wrist on the desk and imagines that there is a QWERTY keyboard within the reachable area of the index finger. Users do not see a keyboard or an image of one on the physical surface that they type on. To reduce the cognitive load of locating a key on the keyboard, the 26 keys were designed to occupy the whole reachable area of the index finger. Figures 1B-E show the recommended boundaries of the imaginary keyboard: the first row of keys lies within reach of finger pulp touches; the third row lies within reach of a fingernail tap; the left and right boundaries are within a comfortable range of movement.

3.1 Layout

Users receive visual feedback from an external display, e.g., an AR/VR headset, a computer monitor or a smart TV. Figure 2 shows the keyboard layout displayed on the output device. The keyboard consists of a QWERTY layout (26 keys) and the candidate selection region. The QWERTY layout reminds the user of the character key locations. During the text input, the system predicts the top-5 probable words and displays them in a ranked



Fig. 2. The keyboard displayed on an external output device. Rank-1 represents the most probable candidate word. The center coordinates of 'Q', 'A', 'Z', 'P' and 'M' are (0,0), (0.2,1), (0.8,2), (9,0) and (6.8,2)

manner: rank-4, rank-2, rank-1, rank-3, and rank-5 candidate words from the left side to the right side on the candidate selection region. That is, the most probable candidate word is in the middle, while the others are on both sides of it. This setting is designed to save time while selecting words. During the text entry, users can focus on the display information instead of the hand movements.

3.2 Usage

Users type on the imaginary keyboard in four steps:

- (1) Tap (touching down and up) for every character of the desired word in an orderly fashion.
- (2) Long press for 200 ms to trigger candidate selection. A pointer will appear on the rank-1 candidate word.
Do not touch up until step four.
- (3) (Optional) If the rank-1 candidate word is not the desired word, swipe left or right to control the cursor.
The cursor move across a candidate word when the IMU ring rotate 7.5° along the vertical axis.
- (4) Touch up to pick the candidate word under the cursor.

Users delete a word by swiping left. Like many other text entry techniques with word suggestions [40, 72, 73], users of QwertyRing type once again if the desired word was not on the candidate list. There is a constraint when using QwertyRing, which is that the IMU can sense the orientation of the index finger, but can not sense the translation of the finger. Users should rest the wrist on a fixed place on the surface and rotate the wrist to touch different keys. Users should not type by moving the whole hand. Only with this constraint, the orientation of the IMU can be distinguishable when the users type on different keys. Users need training to follow this requirement. We will discuss the learning effect in the discussion section.

3.3 QwertyRing Design Rationale

In this subsection, we attempt to justify design decisions of QwertyRing. The following discussions may refer to some results in the second study "DECODER DESIGN", which collected and analyzed users' typing data. Readers can find the details later in the study section.

- (1) *Why use touch gestures (but not buttons) for word selection and deletion?*: The 26 keys occupy the whole reachable area of the index finger, so we used the long press motion for word selection and used the swipe left motion for deletion. This design has two advantages. First, it reduces the user's cognitive load of locating the desired key. For example, there is no button above the 26 keys, so the user can simply straighten the finger to type the key on the first row of the invisible keyboard. Second, it improves the accuracy of word prediction, because there is a tradeoff between the interaction room and the word prediction accuracy [71].
- (2) *Why do we use five candidate words?*: On the one hand, if there are fewer candidate words, the user's desired word is less likely to be on the candidate list. According to the simulation in study two (figure 11), the top-5 accuracy of the text entry decoder was 97.5%, while the top-3 accuracy was only 93%. Users had to type once again if the desired word was not on the candidate list. Though 3-candidate may speed up the word selection phase, 5-candidate is still a more efficient design according to theoretical simulation. On the other hand, if there are more candidate words, users may make mistakes in the word selection phase because of the inaccurate sensing of finger orientation.
- (3) *Why rotate the hand 7.5° to move the cursor across a candidate word?*: In study two, we collected users' typing data when using QwertyRing. The participants were asked to type within a comfortable range of movement. The visualization of users' typing behavior shows that the comfortable range was around 30° (figure 9). Therefore we mapped the five candidate words to the rotations of -15°, -7.5°, 0°, 7.5°, and 15°.
- (4) *Why is the ring on the middle phalanx of the finger? Why is there wrist fixing?*: First, these design choices (constraints) are necessary for the decoder of QwertyRing. QwertyRing is based on the mapping from (Yaw, Pitch) of the ring to (X, Y) on the keyboard. If the ring was on the proximal phalanx, we lost the mapping from Pitch to Y. If the user moved his wrist to type, we lost the mapping from Yaw to X. QwertyRing cannot work with any of these signals missing. Second, these choices are acceptable for users. Prior work [21] shows that the best placement of a ring is the proximal phalanx of the finger, while the second best placement is the middle phalanx. With the wrist fixing, users rotate the wrist to input. Study two shows that it is comfortable to rotate the wrist within 30°.

4 STUDY ONE: TOUCH DOWN/UP SENSING

In this study, we collected motion data from participants' touch on the desk with a finger-worn IMU. The motivation was to design techniques of sensing touch down and touch up on physical surfaces, based on which we can further detect touch events including long press, swipe left and swipe right. Touch down refers to the finger contacting a physical surface. If someone performs a swipe down gesture in the air but does not hit the surface, the system would not report a touch down event. Touch up refers to the finger leaving the physical surface. We explored the recognition of touch events in three steps:

- (1) For **touch down** recognition, we followed prior work [21] to implement a method based on SVM.
- (2) For **touch up** recognition, we analyzed the motion data of touch up and proposed our solution.
- (3) Based on the recognition of touch down and touch up, we sensed long press, swiping left and right by threshold methods.

4.1 Design and Procedure

We recruited twelve participants from a college campus (aged from 19 to 25, M = 21.59, SD = 2.02, 4 females). All the participants were right handed. Figure 3 illustrates the experimental setting. There was a thin rigid touch screen attached to the wooden desk. Participant touched on the touch screen with the IMU ring worn on the middle phalanx of the index finger. The experimental task included touching with different gestures according to the verbal instructions of the researcher. There were four datasets to collect: tapping, long presses, swiping

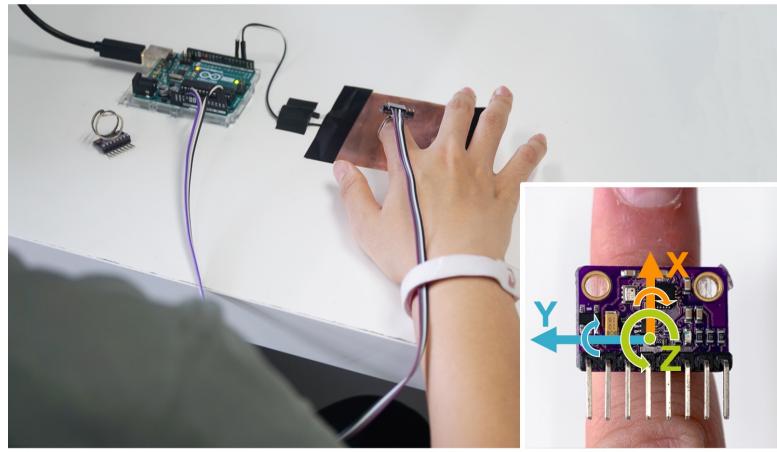


Fig. 3. The experimental setting in study one. The subfigure shows the coordinate of the IMU ring.

and mid-air gestures. The tapping, long presses and swiping datasets were to sample touch down and touch up moments in various conditions. The mid-air gestures served as negative samples of touch.

- (1) **Tapping Dataset:** Participants tapped on the touch screen for $5 \text{ blocks} \times 100 \text{ trials} = 500$ times. They tapped in their preferred way with arbitrary positions, strengths and angles. We collected 6000 tapping samples from the twelve participants in total.
- (2) **Long Press Dataset:** Participants long pressed for $5 \text{ blocks} \times 100 \text{ trials} = 500$ times. They could stay motionless or rub their fingers on the surface during the long press motion.
- (3) **Swiping Dataset:** Participants performed $10 \text{ blocks} \times 100 \text{ trials} = 1000$ times of swiping gesture, including five blocks of swiping left and five blocks of swiping right. Participants swiped as if they were switching the screen of a smartphone. Participants kept their wrists on the desk during the collections of tapping, long presses and swiping.
- (4) **Mid-air Gestures Dataset:** Participants drew mid-air gestures in five blocks of one minute. The gestures were motionless, swiping, drawing circles and squares, trembling and Hololens gestures respectively [21]. Participants were asked to vary the gestures during the data collection. These mid-air gestures served as negative samples of touch. Because "touch" refers to the finger contacting a physical surface, all the possible mid-air gestures belong to negative samples of touch. In this study, we could not enumerate all of the possible mid-air gestures. Instead, we followed prior work [21] to sample mid-air gestures that were specially designed for touch detection.

Participants rested for one minute between two blocks to avoid fatigue. The study was generally completed within one hour.

4.2 Apparatus

As figure 3 shows, the IMU ring was an accelerometer GY-91 attached to a regular finger ring. The IMU connected to an Arduino Uno R3 with Dupont lines. Participants attached the Dupont lines on the wrist with a velcro strap. The ring collected 6-axis motion data including raw acceleration and angular velocity. The raw acceleration data was combined from linear acceleration and gravity. We used Madgwick Filter [37] to split the raw acceleration into linear acceleration and gravity. We did not collect magnetic force from the IMU, because the raw data of the

magnetometer can be affected by devices in the environment. There are some sensor fusion algorithms [37] to stabilize the reading of the magnetometer, but they are not responsive enough (latency > 200 ms) for our project.

The touch screen was a conductive copper sheet connected to the Arduino. The capacitance of the copper sheet will increase if a finger touches on it. We leveraged this phenomenon to judge the contact [5]. The touch screen reported the ground truth of touch condition: whether a finger has contacted the surface or not. The high-speed camera showed that the latency of the touch screen was below 1 ms. The frequencies of the IMU and the touch screen were 1000 Hz. In total, we had eleven dimensions of motion data: timestamp, 3-axis acceleration, 3-axis angular velocity, 3-axis gravity, and the touch condition.

4.3 Recognition

4.3.1 Touch Down. We followed prior work [21], which senses touch down on physical surfaces using a finger-worn IMU. In this paper, we placed the ring on the middle phalanx of the index finger, which was unexplored. We used an IMU with higher frequency and a touch screen with lower latency.

We located touch down moments from the tapping, long presses and swiping datasets according to the ground truth of the touch condition. We cut time windows $[-25ms, 25ms]$ around touch down moments as positive samples. We cut time windows $[-75ms, -25ms]$ as negative samples to avoid reporting the event in advance. Also, we randomly sampled 100 slices from every minute of mid-air gestures as negative samples. Then, we extracted features from the samples as follows: for each dimension of the 9-axis IMU data (acceleration, angular velocity and gravity), we calculated its maximum, minimum, mean, skewness and kurtosis. Then, we concatenated these values to obtain a feature of 45 dimensions and trained an SVM binary classifier.

Leave one out cross-validation shows that the precision of the classifier was 99.8% ($SD=0.2\%$). The recall rate was 99.0% ($SD=1.4\%$). We further packed the classifier into a touch down sensing algorithm: (1) The algorithm does not report touch down if there has been a touch down in the past 50 ms. (2) The algorithm reports touch down only if the classifier detects ten consecutive frames of touch down. The two statements above reduced false positive touch events and reported only one event for each touch down. Then, we simulated the algorithm on the tapping, long press and swiping datasets. The precision was 99.24% ($SD=1.03\%$). The recall rate was 99.37% ($SD=0.94\%$). The average delay was 14.04 ms ($SD=7.40$). We also simulated the negative samples (mid-air gestures). The algorithm reported only 14 false positive touch events during 60 minutes of mid-air gestures.

4.3.2 Touch Up. To sense touch up, it was not suitable to follow the touch down recognition [21]. Touch down can be recognized accurately because of the significant vibration at the touch down moment. However, we found no similar vibration at the touch up moment. Fortunately, the user's finger does not leave the surface before he touches up. Thus, the vertical component of the acceleration A_G is a good feature to detect touch up: A_G is small during the finger tap, but will suddenly increase at the touch up moment. $\cos\theta$ and G_z also help the recognition of touch up, where θ is the angle between acceleration and the vertical axis, G_z is the angular velocity in yaw. Figure 4 shows A_G , $\cos\theta$ and G_z in four typical conditions. These values are easy to recognize at the touch up moment. In addition, the elapsed time from the touch down moment T_{escape} was considered as a feature, because the motion data after touch down was unstable for about 20 ms. T_{escape} can reject the false positive caused by the unstable data. In total, we extracted a feature of 19 dimensions: the escaped time from touch down T_{escape} , the minimum, maximum, mean, skewness and kurtosis of A_G , $\cos\theta$ and G_z .

Then, we sampled training data of touch up detection from the tapping, long press and swiping datasets. Positive samples were time windows $[-25ms, 25ms]$ around touch up moments from the three datasets. We had 24000 positive samples. Negative samples were random time windows of 50 ms between touch down and touch up from the three datasets. We cut three negative samples from each touch, so we had 54000 negative samples. The number of negative samples was larger because it was important to avoid false positives, which could be triggered anytime.

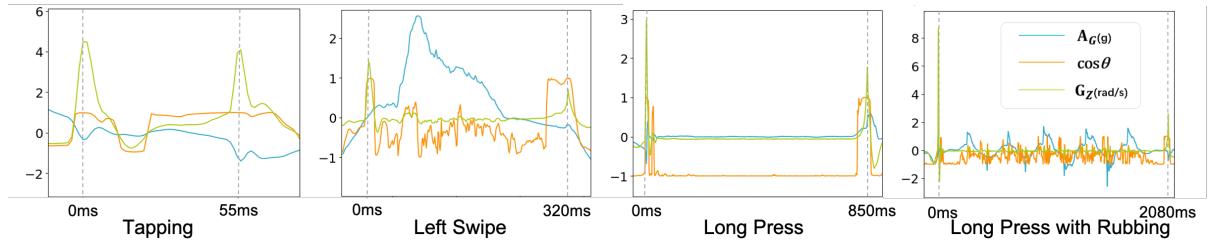


Fig. 4. The motion data between touch down and touch up moments in four typical conditions. The dotted lines indicate the touch down and touch up moments.

Finally, we used SVM to train a binary classifier of touch up. Leave-one-out cross-validation shows a precision of 99.87% and a recall rate of 99.03%. In real use, the system reported touch up only if the classifier detected three consecutive frames of touch up. This statement reduced false positive touch up events. We simulated the algorithm on the tapping, long press and swiping datasets. The recall rates were 98.61% ($SD=1.87\%$), 99.68% ($SD=1.02\%$) and 99.53% ($SD=1.39\%$) respectively. The precision rates were 99.82% ($SD=0.24\%$), 99.78% ($SD=0.31\%$) and 99.65% ($SD=0.42\%$). The delays were 8.55 ms ($SD=4.97$), 7.83 ms ($SD=3.47$) and 7.97 ms ($SD=3.77$).

4.3.3 Long Press. Based on the detection of touch down and touch up, we used a threshold method to recognize the long press gesture: the system reported a long press when the user touched the surface and stayed for more than 200 ms. Staying meant that the touch up event was not triggered and the finger was motionless (the angular velocity was smaller than 0.1 rad/s). The threshold on the angular velocity was an empirical value, which ensured that most swiping gestures would not be misidentified as a long press. To evaluate the performance of this threshold method, we used the long press dataset as positive samples and used the tapping and swiping datasets as negative samples. The recall rate was 98.8%, while the precision was 99.2%. The method also worked without any problems in the following study.

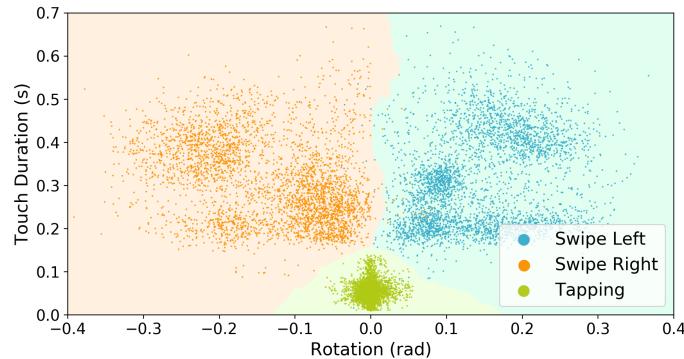


Fig. 5. The comparison of swipe rotation (x-axis) and touch duration (y-axis) among tapping, swipe left and swipe right. The background colors stand for the prediction of the KNN classifier.

4.3.4 Swiping. As we had accurately recognized long press, we only needed to distinguish among tapping as well as swiping left and right. We designed two features to classify them: the touch duration and the rotation (yaw

increment) between touch down and touch up. Figure 5 compared the tapping and swiping left and right with the two features. The touch down and touch up moments here were acquired by the recognition algorithms, but not the ground truth. A simple threshold method can distinguish the three gestures. The method reports swiping left when rotation > 0.02 rad and duration > 100 ms, reports swiping right when rotation < -0.02 rad and duration > 100 ms, and reports tapping otherwise. The accuracy was 97.82%. We trained a ternary KNN classifier ($K = 10$) to further improve the performance. Leave-one-out cross-validation shows an accuracy of 99.38% ($SD=0.43\%$). In real use, the system called the classifier at the touch up moment. It would meanwhile report the swipe left or swipe right movements if the classifier predicted a swiping gesture.

4.4 Touch Sensing on Different Surfaces

In the aforementioned study, we evaluated our touch sensing algorithm in detecting touches on a wooden desk. We collected motion data from participants' touches on a wooden desk with the finger-worn IMU. There was a thin copper sheet attached to the desk which served as a touch screen to collect the ground truth of the touch condition. However, with the copper sheet, we could not get the perfect motion data of touching on the desk. Further, we need to evaluate our algorithm on different surfaces.

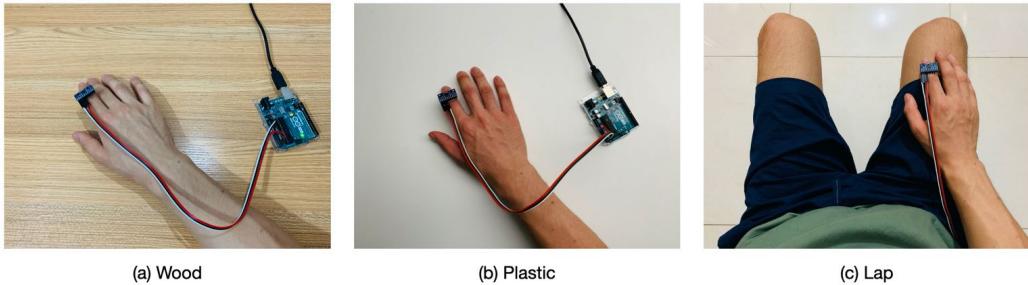


Fig. 6. The experimental setting.

We conducted an external experiment to evaluate our algorithm on other physical surfaces, including a wooden desk (without the copper sheet), a plastic desk, and the top of a human lap (figure 6). Twelve participants took part in the experiment (aged from 20 to 26, $M = 22.03$, $SD = 3.37$, 3 females). They did not attend the last study. The experimental settings are the same as the last study except (1) we collected the tapping, long presses and swiping datasets on three types of surfaces; (2) there was no copper sheet (touch screen) attached to the surface; (3) the experiment lasted for three hours, so the participants completed this experiment in three days to avoid fatigue; (4) the participants pressed the enter key to continue after every five touches.

The challenge of this experiment was that we could not get the ground truth of the touch condition via a touch screen. To overcome this problem, we learned the pattern of data and labeled the touch down and touch up events manually. Because the participants pressed the enter key to confirm every five touches, it was easy to locate the touch moments manually in each segment of data. We were confident that the data was correctly labeled. We used all the data of the three surfaces to train general models of touch down and touch up sensing. The two models were supposed to work on different surfaces. We adopted the same methods (the same SVM features) as the first study to train the models. Then, we used leave-one-out cross-validation to evaluate their performances through simulations on the tapping, long press and swiping dataset.

Table 1 shows the result. For touch down detection, RM-ANOVA showed a significant effect of surface on the recall rate ($F_{2,22} = 25.14$, $p < .001$, $\eta_p^2 = 0.696$). Bonferroni-corrected post-hoc tests showed significant differences

	Touch Down		Touch UP	
	Recall	Precision	Recall	Precision
Wood	99.54% (SD=0.82%)	99.45% (SD=0.46%)	99.45% (SD=0.58%)	99.65% (SD=0.23%)
Plastic	99.67% (SD=0.82%)	99.22% (SD=0.46%)	99.38% (SD=0.68%)	99.71% (SD=0.23%)
Lap	97.69% (SD=1.42%)	99.32% (SD=0.76%)	99.18% (SD=0.68%)	99.66% (SD=0.15%)

Table 1. The accuracy of touch down and touch up sensing on different surfaces.

between the following surface pairs: Wood-Lap ($p<.005$) and Plastic-Lap ($p<.005$). The recall rate of sensing touch down on the lap is significantly lower (97.69%). There is no significant effect of surface on the precision rate. That is, the touch down sensing algorithm performs well on rigid surfaces (the wooden desk and the plastic desk). The algorithm missed some touches when the users tapped on the lap surface. For touch up detection, there is no significant effect of surface on either recall rate nor precision rate. That is, the touch up sensing part worked well on detecting touches on all the tested surfaces. We asked the participants to try our algorithm with trained models. Participants reported that they cannot challenge the algorithm when touching on rigid surfaces. Participants could deliberately perform unrecognized touches on the lap by touching lightly, but the algorithm worked well if they touched with normal strength.

4.5 Lower Frequency

We used 1000 Hz as a frequency to evaluate the ceiling performance of touch sensing. However, a practical system may be more optimized by using a lower frequency to save battery life. However, there is a tradeoff between frequency and sensing accuracy. We evaluated the performance of touch sensing with downsampling (500 Hz, 200 Hz and 100 Hz) to reveal this tradeoff. Figure 7 shows the result in F_1 score, which is the harmonic mean of the precision and recall.

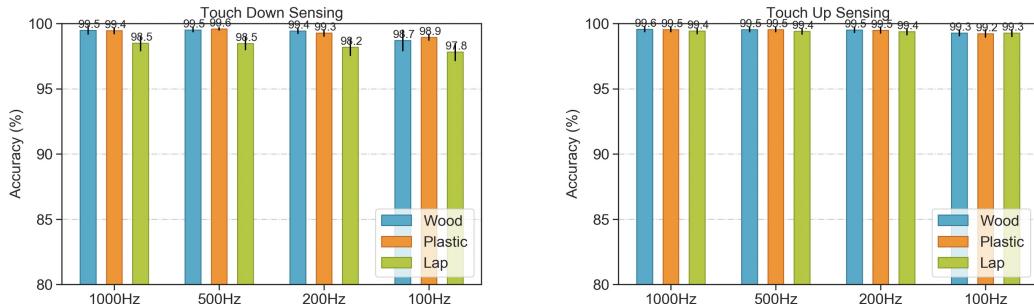


Fig. 7. The accuracies of touch down and touch up sensing over sample frequency. Error bars indicate 95% confidence interval.

For touch down sensing, two-factor RM-ANOVA showed a significant effect of frequency on the performance ($F_{3,33} = 16.44, p < .001, \eta_p^2 = 0.599$). Bonferroni-corrected post-hoc tests showed significant differences between the following surface pairs: 1000Hz-100Hz($p<.001$), 500Hz-100Hz($p<.001$), 200Hz-100Hz($p<.05$). For touch up sensing, there was also a significant effect of frequency on the performance ($F_{3,33} = 39.94, p < .001, \eta_p^2 = 0.0.784$).

Post-hoc tests showed significant differences between the following surface pairs: 1000Hz-100Hz($p < .001$), 500Hz-100Hz($p < .001$), 200Hz-100Hz($p < .001$). Results show that there is a tradeoff between the frequency and the performance, but the accuracy was still high at 200 Hz.

5 STUDY TWO: DECODER DESIGN

The motivation of this study was to obtain users' typing data to design the text entry decoder. When typing with QwertyRing, users cannot see the keyboard layout on the physical surface. Instead, users type on an imaginary keyboard according to the muscle memory of their index fingers. In this study, we determined the spatial parameters of the imaginary keyboard from users' typing data. We also wanted to investigate users' typing behavior.

5.1 Design and Procedure

We recruited twelve participants from campus (aged from 19 to 29, $M = 23.67$, $SD = 3.14$, 4 females). They did not take part in the first study. There was a training phase before the experiment. We introduced the concept of an imaginary keyboard. We showed the participants the recommended boundaries of the imaginary keyboard (figure 1B ~1E). Then, participants typed 'A' to 'Z' orderly twice to get familiar with the position of each key in the imaginary keyboard. It was a common mistake to input text by moving the hand instead of the finger. The researcher corrected the participant in this case. The training phase lasted for about five minutes. The experiment consisted of five repeated sessions. In each session, participants transcribed ten phrases, which were randomly sampled from the MacKenzie phrase set [36].



Fig. 8. The experimental setting of study two. A participant touched on a desk to input.

As figure 8 shows, the participant sat on an adjustable chair. He was able to adjust the chair to a comfortable position. An external monitor displayed the user interface, which consisted of the QwertyRing keyboard layout,

a task phrase, and the inputting phrase. Participants wore the IMU ring and typed on a regular desk. They were asked to transcribe the task phrase as fast and accurately as possible. We did not have a decoding algorithm at this stage. The system always displayed the correct character no matter how the participant typed. However, the user should redo the phrase if he found a mistake by himself. A similar experimental design was used to collect ideal typing data in prior work [4, 14, 35]. Participants were asked to focus on the display information but not their hand movement. The study lasted for one hour.

5.2 Processing

The basic idea of QwertyRing is to predict users' desired keys by the orientation of the IMU. We inferred that the pitch angle and the yaw angle of the IMU is useful for decoding: (1) The pitch angle is the angle between the forward direction and the horizontal plane. When the user taps on different rows of keys on the imaginary keyboard, the pitch angles of the IMU are different. (2) The yaw angle is the rotation around the vertical axis. When the user taps on different columns of keys on the keyboard, the yaw angles are different. We used Madgwick Filter [37] to obtain the pitch angle of the IMU. However, we can not acquire the yaw angle accurately because of the drifting issue of the IMU. Instead, we estimated the increment of yaw between two taps (ΔYaw) by the integration of the angular velocity in yaw:

$$\Delta Yaw = \sum_{i=1}^T G_{zi} \Delta t_i \quad (1)$$

where T is the number of frames between two taps, G_{zi} is the z-axis of gyroscope data in the i th frame, Δt_i is the duration of the i th frame.

We formulate the concerned data. Suppose that participants typed on the key u for N_u times in the experiment, we denote the pitch angles of the IMU as $P_u = \{Pitch_{ui}\}_{i=1}^{N_u}$ and the yaw angles of the IMU as $Y_u = \{Yaw_{ui}\}_{i=1}^{N_u}$. Suppose that participants continuously typed on the keys u and v for $N_{u,v}$ times, we use $\Delta Y_{u,v} = \{\Delta Yaw_{u,vi}\}_{i=1}^{N_{u,v}}$ to represent the yaw increments. Limited by the sensing ability, the 6-axis IMU only collected the absolute pitch angles P_u and the relative yaw angles $\Delta Y_{u,v}$, but no absolute yaw angles Y_u . For each P_u and $\Delta Y_{u,v}$ of each participant, errors greater than three standard deviations from the mean were removed.

5.3 Results

Users typed on each key u with similar finger orientations. Thus, P_u and Y_u obey distributions. According to the relative model in BlindType [35], $\Delta Y_{u,v}$ also obey a distribution. To simply computation, we assume that P_u and $\Delta Y_{u,v}$ obey two individual normal distributions. Similar assumption was made in prior work [35, 73]. Because P_u and $\Delta Y_{u,v}$ were known, we fitted their distributions $(\bar{P}_u, \sigma P_u)$ and $(\bar{Y}_{u,v}, \sigma Y_{u,v})$. These distributions were used in the text entry decoder, which will be introduced in the next section.

Y_u was unknown. We estimated its mean value to understand the imaginary keyboards in the users' minds. In detail, we sought all \bar{Y}_u to minimize the formula as follow:

$$\sum_u \sum_v N_{u,v} ((\bar{Y}_v - \bar{Y}_u) - \bar{\Delta Y}_{u,v})^2 \quad (2)$$

where we used the steepest descent algorithm to reach minimization.

Figure 9 shows the point cloud (\bar{Y}_u, \bar{P}_u) of all participants. As expected, the yaw and pitch angles can locate inputted keys in X and Y coordinates respectively. The left and right boundaries of the keyboard lie on 'Z' and 'P'. The gap in yaw is 32.4° . The upper and lower boundaries of the keyboard lie on 'Q' and 'M'. Their pitch angles are 24.1° and 62.6° . The gap is 38.4° .

As figure 10 shows, There is a large user difference on the size and the shape of imaginary keyboards. The biggest keyboard is $40.1^\circ \times 45.7^\circ$ in yaw and pitch (P3), while the smallest is $25.3^\circ \times 33.4^\circ$ (P4). P1 typed the

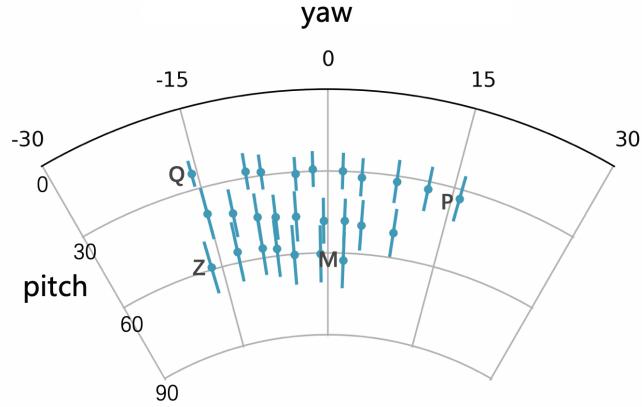


Fig. 9. The point cloud of all the participants. Error bars indicate standard deviation of the pitch angles.

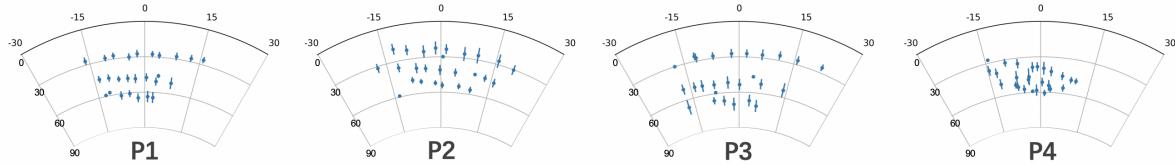


Fig. 10. The point clouds of four typical participants. Error bars indicate standard deviation of the pitch angles.

keys in the same row with similar pitch angles. That is, the keyboard in his mind was fan-shaped. P2 tended to straighten the finger when typing keys in the upper left corner. We infer that his imaginary keyboard was a rectangle biased to the left side so that keys in the upper left corner were the farthest.

6 QWERTYRING DECODER

We used the Bayesian approach to develop the QwertyRing decoder. The approach calculates the probability as follow:

$$P(W|I) \propto P(I|W) \times P(W) \quad (3)$$

where W is a candidate word in the dictionary and I is a sequence of input touchpoints.

We discuss the design of the decoder with the touch model and the language model. The evaluation in this section is based on the data in the second study.

6.1 Touch Model

$P(I|W)$ is the touch model part of the bayesian decoder:

$$P(I|W) = \prod_{i=1}^n P(I_i|W_i) \quad (4)$$

where n is the length of W . I_i and W_i indicate the i th touch point and the i th character respectively.

In practice, researchers calculated $P(I_i|W_i)$ using a bivariate normal distribution [4, 17]. Prior work found that vectors between successive touch points also obey a normal distribution [35]. In this paper, we assumed that the absolute pitch angles P_u and the relative yaw angles $\Delta Y_{u,v}$ obey two individual normal distributions. Thus, $P(I|W)$ was estimated as follow:

$$P(I|W) = \prod_{i=1}^n P(Pitch_i|W_i) \prod_{i=1}^{n-1} P(\Delta Yaw_{i,i+1}|W_{i,i+1}) \quad (5)$$

where $Pitch_i$ is the pitch angle of the IMU when the participant types W_i , $\Delta Yaw_{i,i+1}$ is the increment of yaw angle between W_i and W_{i+1} .

The decoder calculates $P(Pitch_i|W_i)$ and $P(\Delta Yaw_{i,i+1}|W_{i,i+1})$ with the normal distributions of P_{W_i} and $Y_{W_i, W_{i+1}}$ introduced in study two.

6.2 Language Model

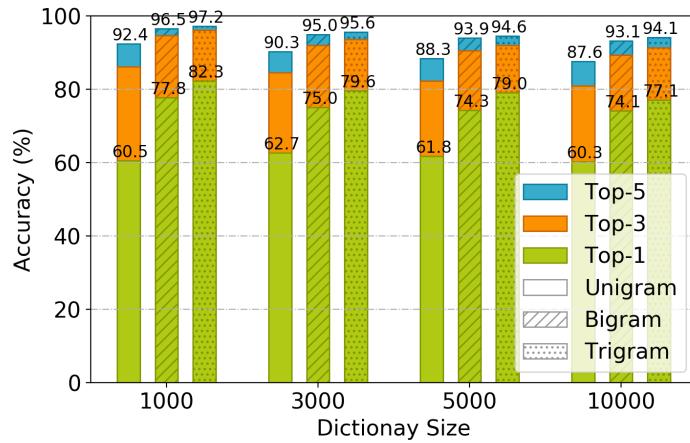


Fig. 11. The top-1, top-3 and top-5 accuracies of different language models over dictionary sizes.

$P(W)$ is the language model part. We tested unigram, bigram and trigram models [28] by simulation. Figure 11 shows the prediction accuracy of each model over dictionary sizes. The top-1, top-3 and top-5 accuracies mean the probability of predicting users' desired word as the best match, one of the top-3 and top-5 matches respectively.

Two-factor RM-ANOVA shows a significant effect of language model on top-1 accuracy ($F_{2,22} = 86.53, p < .001$). The trigram model is better than unigram ($p < .001$) and bigram ($p < .001$). It achieves a top-1 accuracy of 79.0% and top-5 accuracy of 94.6% in a dictionary of 5000 words. We used the trigram model in the remainder of this paper. The effect of dictionary size on top-1 accuracy is only a trend ($F_{3,33} = 2.41, p = .085$). We speculate that this effect occurs because the experimental tasks contain few low-frequency words.

6.3 Personalization

The user difference found in the second study inspired us to explore the personalization of the touch model.

- (1) The **general** model fits the touch model by all the data in the second study.
- (2) The **personal** model is the same as the general model at first. The personal model updates itself continuously by processing personal typing data.

In practice, the user can only use the general model at the beginning. The system collects personal data and tries to update some distributions of P_u and $\Delta Y_{u,v}$ to obtain the personal model. In detail, when the size of P_u or $\Delta Y_{u,v}$ of the user is large enough ($>=8$), the system replaces the corresponding distribution using personal data. The threshold ($>=8$) performed the best in simulation.

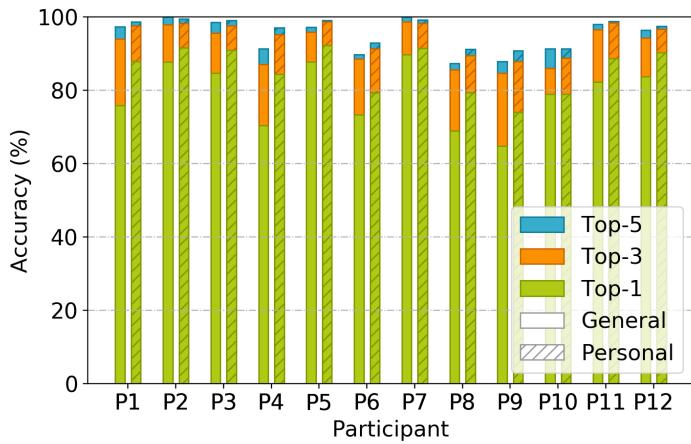


Fig. 12. The prediction accuracies of the general model and the personal model among participants.

Figure 12 compares the performance of the two models by simulation. We used 5-fold cross-validation to evaluate the personal model. That is, we trained the personal model by four sessions of data (40 transcribed phrases) and tested on the other 10 phrases. The average accuracy of the personal model is higher. We further compared the two models through a third study.

7 STUDY THREE: QWERTYRING FIVE-DAY STUDY

The goal of this study was two-fold: first, to evaluate the input rate and the learning effect of QwertyRing text entry; second, to compare the performance of the general model and the personal model.

7.1 Design and Procedure

We conducted a five-day study to evaluate the performance of QwertyRing. The study followed a between-subject design to compare the general model and the personal model. We recruited 16 participants from a college campus (aged from 20 to 27, $M = 22.86$, $SD = 2.09$, 6 females). They did not take part in the previous studies. On each day of the study, participants transcribed 30 phrases. The phrases were randomly sampled from the MacKenzie phrase set [36].

The experimental apparatus were similar to study 2 (figure 8). Participants sat on an adjustable chair, wore the IMU ring and typed on a regular desk. They gained visual feedback from an external monitor display. Participants were asked to transcribe the task phrases as fast as possible. During the text inputting, participants were asked to focus on the display information but not their hand movements.

All of the participants used the general model on the first day. The general model used all the data in the second study to fit the touch model. After the first day, we divided the participants into two groups. Each group had eight people. The average ages of the two groups were 22.50 ($SD = 1.77$) and 23.25 ($SD = 2.55$). The two groups were well-matched in typing speed. In detail, we ran a program to divide the participants randomly for 1000 times and picked the two groups of which the average performances were closest. The first group of participants continued to use the general model in the following four-day study. The second group used the personal model instead. For each participant of the personal model group, we customized the personal touch model based on existing data from each participant every day.

There was a training phase before the experiment on day 1. The researcher introduced the concept of imaginary keyboard. With the guidance of the researcher, participants typed 'A' to 'Z' in order twice to become familiar with the keyboard. Then, they tried to input five phrases. The training phase lasted for ten minutes in total. For each day of the study, participants transcribed 30 phrases in three sessions. They rested for five minutes between two sessions. On average, participants spent half an hour each day.

7.2 Result

A mixed ANOVA was conducted for text entry speed, Uncorrected Error Rate (UER) and Corrected Error Rate (CER) with model as the between factor and day as the within factor. As UER and CER violated the normalcy, we used the Aligned Rank Transform [67] for correction. If any independent variable or combinations had significant effects ($p < 0.05$), we used Bonferroni-corrected post-hoc tests for pairwise comparisons.

7.2.1 Speed. Text entry speed is measured in Words Per Minute (WPM) with this formula [2]:

$$WPM = \frac{|S| - 1}{T} \times 60 \times \frac{1}{5} \quad (6)$$

where $|S|$ is the length of the transcribed string (including blank spaces), and T is the complete time.

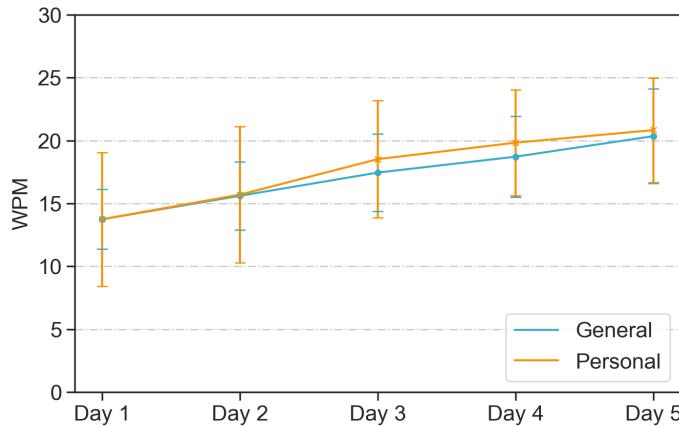


Fig. 13. Text-entry rates of the general model and the personal model over days. Error bars indicate 95% confidence interval.

Figure 13 shows the speeds over five days. The general model starts with a speed of 13.75 WPM ($SD = 2.65$) on day 1 and ends with a speed of 20.35 WPM ($SD = 4.20$) on day 5. The personal model starts with

13.74 WPM (SD = 5.33) and ends with 20.83 WPM (SD = 4.14). There is no significant effect of model on speed ($F_{1,14} = 0.09, p = 0.77, \eta_p^2 = 0.006$). Day has a significant effect on speed for both the general model ($F_{4,28} = 27.00, p < .001, \eta_p^2 = 0.794$) and the personal model ($F_{4,28} = 41.17, p < .001, \eta_p^2 = 0.855$). For the general model, pair-wise comparisons showed significant differences between the following day pairs: 1-3($p < .05$), 1-4($p < .005$), 1-5($p < .005$), 2-4($p < .05$), 2-5($p < .05$), 3-5($p < .05$), and 4-5($p < .05$). For the personal model: 1-3($p < .01$), 1-4($p < .005$), 1-5($p < .001$), 2-3($p < .005$), 2-4($p < .005$), 2-5($p < .005$), 3-5($p < .05$). The interaction effect between model and day was not significant ($F_{4,56} = 0.60, p = 0.66, \eta_p^2 = 0.041$). The learning curve seemed not to converge at the end of the study, so the result does not reflect the ceiling rate.

7.2.2 Error Rate. Two metrics were used to measure text entry accuracy: (1) Uncorrected Error Rate (UER) - text entry errors which remain in the transcribed string. UER is the number of uncorrected erroneous words divided by the number of correct and erroneous words. (2) Corrected Error Rate (CER) - text entry errors which are fixed (e.g., backspaced) during entry. CER is calculated by the number of corrected erroneous words divided by the number of correct and erroneous words. As UER and VER violated the normalcy, we used the Aligned Rank Transform for nonparametric factorial analysis [67].

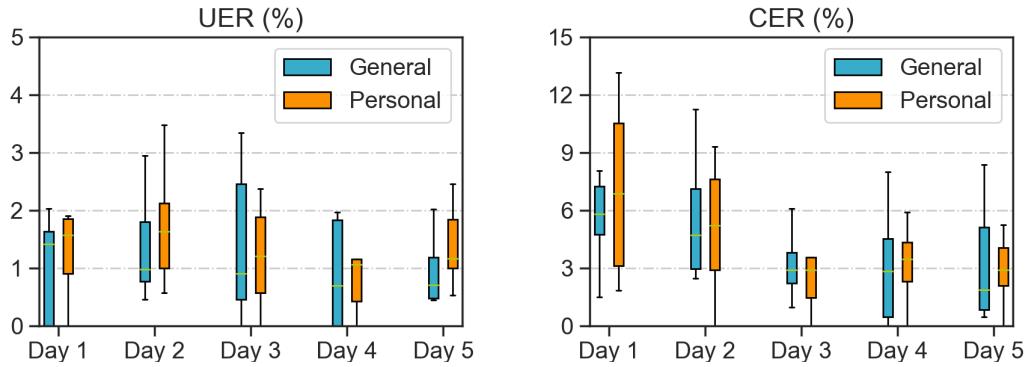


Fig. 14. Uncorrected error rates and Corrected error rates of the two models over days.

Figure 14 shows the UER and the CER over five days. There is no significant effect of model or day on UER. The average UER was 1.17% (SD = 1.02%) for the general model and 1.50% (SD = 1.40%) for the personal model. There is no significant effect of the model on CER. The days have a significant effect on CER ($F_{4,56} = 6.84, p < .005$). Pair-wise comparisons showed significant differences between the following day pairs: 1-3($p < .005$), 1-4($p < .05$), 1-5($p < .005$), 2-3($p < .05$) and 2-5($p < .05$). On day 5, the average CER was 3.22% (SD = 2.92%) for the general model and 2.92% (SD = 1.65%) for the personal model. That is, participants made corrections once every 30 words with both of the models.

7.2.3 Decoder Performance. Figure 15 shows the prediction accuracy of the two models over days. Mixed ANOVA shows no significant effect of model on top-1, top-3 and top-5 accuracy. The days have no significant effect on accuracy for the general model. For the personal model, day has a significant effect on top-1 accuracy ($F_{4,28} = 3.45, p < .05, \eta_p^2 = 0.330$). The result shows that the general model did not improve over days, while the personal model improved. Mixed ANOVA shows a trend that the personal model surpassed the general model on top-1 accuracy from the third day on ($F_{1,14} = 3.22, p = 0.09, \eta_p^2 = 0.187$).

7.2.4 Time Components. To gain deeper insight into performance improvement, we break down the text entry time into three components: typing time, selecting time and pause time. Typing time is the time spent on typing

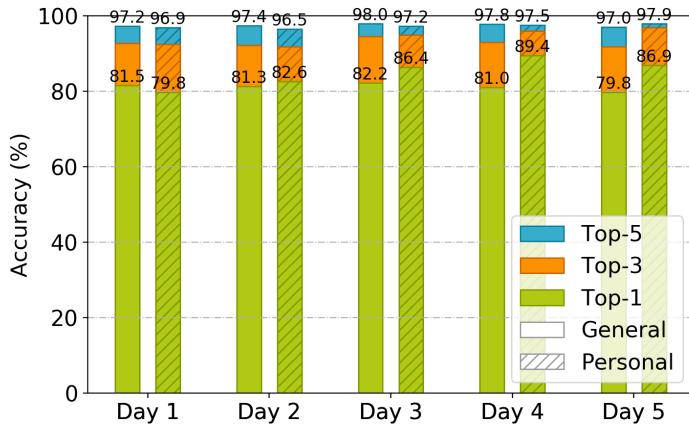


Fig. 15. The top-1, top-3 and top-5 accuracies of the two models over days.

keys. It is calculated by the time from typing the first key to typing the last key. Selecting time refers to the time from typing the last key to selecting a candidate word. Pause time refers to the time from inputting a word to starting to type another word.

Figure 16 shows the time components of all the participants over days. ANOVA shows significant effects of day on typing time ($F_{4,52} = 8.01, p < .001, \eta_p^2 = 0.381$), selecting time ($F_{4,52} = 24.80, p < .001, \eta_p^2 = 0.656$) and pause time ($F_{4,52} = 18.86, p < .001, \eta_p^2 = 0.592$). There is no significant effect of the model on these time components. The result shows that users learned to type and select faster without sacrificing accuracy.

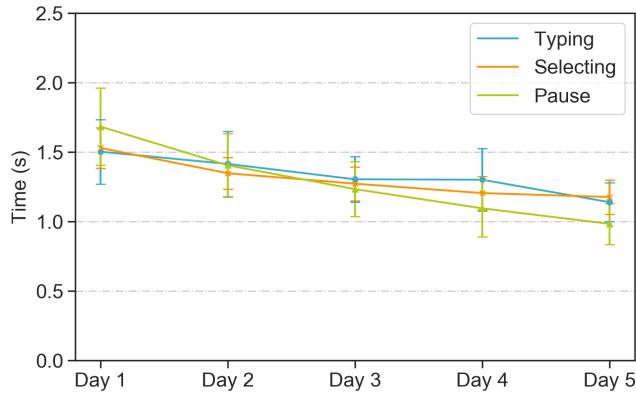


Fig. 16. The time components of inputting a word over days. Error bars indicate 95% confidence interval.

We further analyze the selecting time when the desired word was predicted as different ranks. Figure 17 shows the selecting time over ranks. The bars in the figure are in the same order on rank as the candidate selection

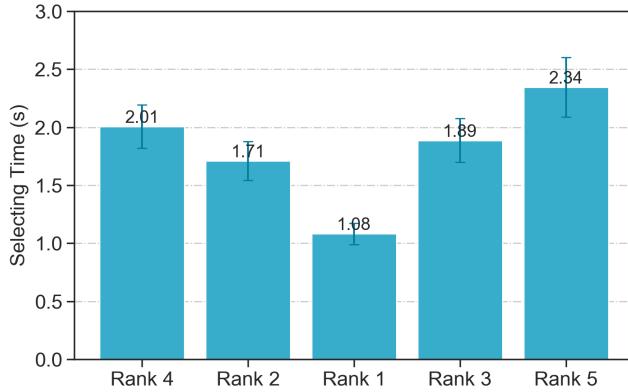


Fig. 17. The selecting time over prediction ranks. Error bars indicate 95% confidence interval.

region. Participants selected the rank-1 candidate the fastest because the pointer was already placed on it at the beginning. For other candidates, we found two factors that significantly affect the selecting time. The first influential factor is the distance to the middle ($F_{1,15} = 69.47, p < .001, \eta_p^2 = 0.822$). The second influential factor relies on which side of the middle the word is placed on ($F_{1,15} = 39.76, p < .001, \eta_p^2 = 0.726$). The factor is influential due to the asymmetric movement constraints of the wrist [19]. The results support our design of the candidate selection region, which places the most-probable word in the middle, the rank-2 and rank-4 words in the left side, and the rank-3 and rank-5 words in the right side.

7.2.5 Subjective Rating and Feedback. Participants rated the subjective speed, accuracy, fatigue, and preference on a 7-point Likert scale (1 - the worst; 7 - the best) on day 1 and day 5. The ratings are acceptable on day 1 and improves by day 5. For the general model, a Wilcoxon Signed-Rank test shows that the subjective speed ($Z = -2.56, p < .05$) and preference ($Z = -2.07, p < .05$) significantly improves on day 5. For the personal model, the subjective speed ($Z = -2.21, p < .05$), accuracy ($Z = -2.41, p < .05$) and preference ($Z = -2.41, p < .05$) significantly improves on day 5. There is no significant effect of the model on all the measurements.

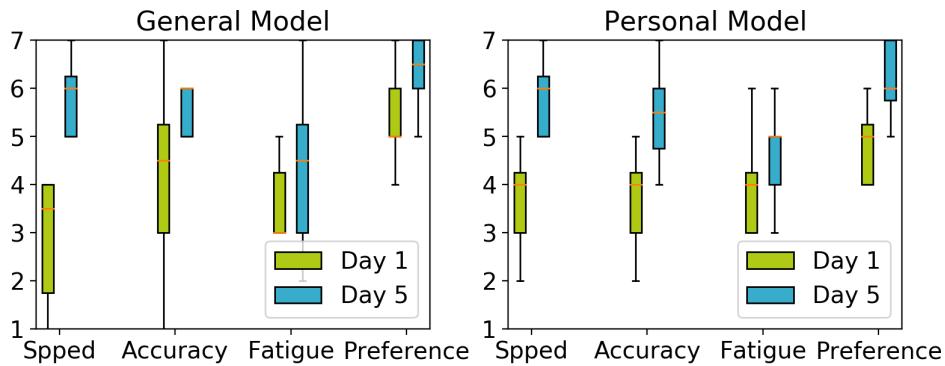


Fig. 18. Subjective ratings (higher is better).

Two participants reported fatigue after 20 minutes of typing on day 1. However, they felt better over days. The imaginary keyboards of these fatigued participants were too large. When they learned that the keyboard could be smaller, they felt the issue of fatigue problem was reduced. The participant with the best performance reached 19.37 WPM on day 1 and 29.12 WPM on day 5. He followed the requirement of typing from the first day on. The participant with the slowest speed reached 7.54 WPM on day 1 and 16.67 WPM on day 5. Though being trained on day 1, he still moved his whole hand instead of rotating the wrist to type. QwertyRing could not decode his desired words well. However, he performed better over the study days and finally reached an acceptable input rate.

7.3 Comparison to Index Finger Typing on Smartphones

We conducted an external experiment to evaluate index finger typing on smartphones. The motivation was to compare QwertyRing against a commonly use keyboard. We chose index finger typing on smartphones as a benchmark because (1) users are familiar with smartphone text entry; (2) both QwertyRing and this setting use index finger typing. The same 16 participants of the third study took part in the experiment. The task was to transcribe 30 phrases from the same phrase set as in the third study. Participants used their own smartphones and inputted the 30 phrases in three blocks. They were encouraged to take a 1-minute break between two blocks. Before the experiment, participants had ten minutes to practice. The experiment lasted for about 20 minutes.

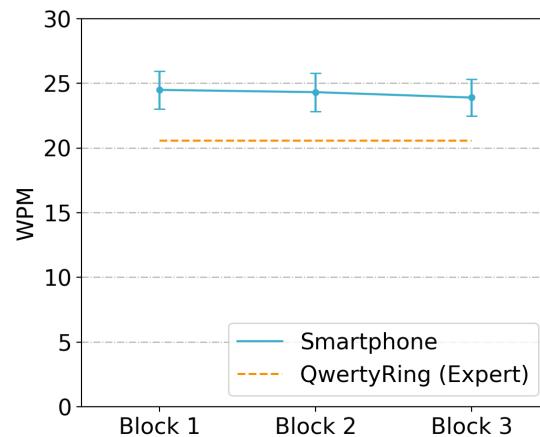


Fig. 19. The blue line shows the speed of index finger typing on smartphones. Error bars indicate 95% confidence interval. The orange dashed line is the expert speed of QwertyRing, which is 86.48% of the blue line.

Figure 19 shows the typing speed over blocks (blue), compared with the expert speed of QwertyRing (orange). Participants reached an average speed of 23.81 WPM in the three blocks. ANOVA shows no significant effect of block on speed ($F_{2,30} = 0.20, p = .82, \eta_p^2 = 0.013$), that is, participants might reach their ceiling typing speeds. The performance of QwertyRing was 86.48% of this result. It shows that QwertyRing provides a wearable text entry approach without negatively affecting the performance. We acknowledge that this external experiment is informal, e.g., the duration was shorter than typing with QwertyRing. However, the result of this experiment represents the typing ability of the participants and helps us to understand how well users perform on QwertyRing, relative to a commonly use keyboard.

8 DISCUSSION

8.1 Comparison to Other Techniques

With ten minutes of practice, participants reached a speed of 13.75 WPM, with a UER of 1.5%. After five days of training, participants reached an expert speed of 20.59 WPM, with a UER of 1.3%. This outperforms existing text entry techniques using a single ring. The upper half of table 2 compares QwertyRing with existing ring-based text entry techniques. The novice metrics indicate the performance in the first session of evaluation, while expert metrics indicate the best performance in the last session. QwertyRing and *RotoSwype* conducted evaluations in five days. *ThumbText* and *TipText* conducted multi-session experiments with at least 40-minute of typing. QwertyRing has the best performance on both the novice speed and the expert speed. The lower half of table 2 shows the average typing speeds on smartphones and smartwatch. The performance of QwertyRing is close to the speed of typing on a smartwatch, and is two thirds of the speed of smartphone typing.

	Novice WPM	Expert WPM	Novice UER%	Expert UER%
QwertyRing (General)	13.75	20.35	1.03	0.95
QwertyRing (Personal)	13.74	20.83	1.90	1.63
RotoSwype (Ring) [24]	9.2	14.8	2.4	0.5
ThumbText (Ring) [30]	5.5	11.4	13.3	9.1
TypingRing (Ring) [45]	NA	8.2	NA	NA
TipText (Ring) [68]	10.5	13.3	0.3	0.3
Smartphone [54]	30.3		2.4	
Smartwatch [18]	22.0		1.5	

Table 2. The comparison between QwertyRing and relevant techniques. We acknowledge that the comparison is not absolutely fair because the user expertise in different studies varies greatly.

8.2 The Learning Effect

Users of QwertyRing should rest the wrist on a fixed place and rotate the wrist to type. Users need training to follow this requirement. In study three, there was a training phase of ten minutes: the researcher introduced the concept of QwertyRing. Then the participants followed the instructions to practice. After the ten-minute training, participants typed with a speed of 13.74 WPM. This WPM demonstrated that QwertyRing is easy to start using. In commercial product, a smartphone application can help users get started. After the five-day study of 30 minutes, participants reached a speed of 20.59 WPM. The learning curve seems not to converge. It shows that the expert speed of QwertyRing is fast.

8.3 The General Model or the Personal Model?

We found no significant effect of the model on speed, accuracy, decoder performance, and subjective feedback. The result remains the same if we exclude the data on day 1, where the personal model has not been trained. We give two reasons. First, the general model already decoded well. There is less room for the personal model to improve. Second, the improvement of decoding was not crucial. Participants inputted text faster mainly because they typed and selected the correct words faster. However, we still recommend the personal model for the current prototype. The result of the third study shows a trend that the personal model decodes better than the general model from the third day on ($p = 0.09$). According to the subjective feedback, participants could feel the improvement of the personal decoder over days, while they found no improvement of the general decoder.

8.4 What If the Ring is Worn on the Proximal Phalanx?

As we have discussed before, we placed the ring on the middle phalanx to optimize the performance of the text entry decoder. However, the proximal phalanx is a placement with better social acceptance [21].

Unfortunately, QwertyRing cannot support text entry if the ring is worn on the proximal phalanx. Specifically, this setting cannot distinguish between the characters on the second row of the keyboard and those on the third row, because the motion data is very closed in these two situations. We invited four expert participants (top-4) in study three to try it, with the personal model updated. They could not finish a sentence as some of the desired words were not in the candidate list.

The four participants could enter words when we reduced the dictionary size to 100 words. In this situation, the Bayesian decoder could find the desired word because the possible words were few enough. Thus, it is a good idea to have QwertyRing worn on the proximal phalanx in some use scenarios. Users can trigger shortcuts on a smartphone by entering a command with QwertyRing, for example, “music” for playing music.

8.5 What If Typing on Other Surfaces?

The decoding performance of QwertyRing is affected by two factors: (1) touch sensing accuracy, and (2) the motion data at touch moments. We argue that QwertyRing supports text entry on any desk-like surfaces, which are rigid, flat and spacious enough for a keyboard. There are two reasons. First, touch sensing works well on rigid surfaces, such as wooden and plastic desks. Second, the motion data at touch moments is supposed to be very similar when users type on different flat and spacious enough surfaces.

However, we did not think that QwertyRing could support text entry on the leg, because the top of the leg is not flat and spacious enough, which significantly affects the motion data of the finger ring. To verify our hypothesis, we invited four expert participants (top-4) in study three to attend an informal study. The study evaluated the performance of text entry on the wooden desk, the plastic desk and the top of the lap. The four expert participants transcribed ten sentences with a speed of 24.68 WPM ($SD = 4.13$) and 23.96 WPM ($SD = 2.42$) on the wooden desk and the plastic desk respectively. However, the users could not finish all the tasks on the top of thighs, as some of the desired words were not in the candidate list. Thus, we acknowledge that QwertyRing only supports text entry on desk-like surfaces. QwertyRing enables touch down and touch up sensing on the lap, but does not support text entry on the lap.

8.6 The Battery Life

In this paper, we used 1000 Hz as a frequency to evaluate the ceiling performance of QwertyRing. We acknowledge that sending the IMU data at 1000 Hz to a host PC with a wireless connection would drain a ring size battery fast. There are several approaches to save the battery life. (1) First, QwertyRing can work with a simple-rate of 200 Hz. The simulation in the first study showed that the touch sensing algorithm worked well at 200 Hz. A sample-rate of 200 Hz is also enough for the text entry decoder, which leverages low frequency signals (Pitch and Yaw). (2) Second, we can add a physical switch on the ring. Users can turn off the ring when not using it. Ideally, we should develop an algorithm to turn on and turn off the ring seamlessly. (3) Third, as the prediction phase of a SVM model is not high computation consumption, we can detect touch events on the ring and send the results to the host device.

The above approaches will mitigate the power consumption issue. Oura Ring (<https://ouraring.com/>) is an example, which contains more sensors including infrared optical pulse measurement, IMU, body temperature sensor, Bluetooth, and battery. The battery of Oura Ring is 4-6 grams in weight, 7.9mm wide and 2.55mm thick. The battery life is up to one week. Furthermore, we believe that improvements in hardware will eventually solve the problem of power consumption.

9 LIMITATION AND FUTURE WORK

This research has a number of limitations, which suggest new directions for future work:

- (1) The decoder of text entry can be improved. We used a simple key-based touch model because of the user difference and the limited number of participants. More sophisticated language models such as Lattice LSTM [36] and Bert [12] may further improve the performance. We acknowledge that the obtained performance does not reflect the ceiling rate of QwertyRing.
- (2) QwertyRing supports text entry only on desk-like surfaces currently. In the future, it is valuable for QwertyRing to support text entry on vertical surfaces and typing on the body, for example, such as the lap and the palm of the non-dominant hand.
- (3) The IMU rings in our experiments were wired and not small enough, which may affect the user performance. In the experiment, we attached the wires on the user's wrist with a velcro strap to reduce the influence. In the future, it is valuable and possible to develop a wireless IMU ring.
- (4) In daily use, clapping or using everyday objects might lead to unintentional touches. Researchers can experiment with a physical button on the ring as the on-off switch.
- (5) Like many text entry techniques with inaccurate input [16, 18, 24, 40], QwertyRing does not support OOV (Out of Vocabulary) words. To enter punctuations and numbers, we can add a swipe down gesture to switch to punctuation/number mode.
- (6) We did not evaluate QwertyRing in a VR setting. VR text entry is a potential use scenario of QwertyRing, because (1) VR helmets can render the layout as normal displays do, and (2) QwertyRing requires no visual attention to the hand, which is specially designed for VR text entry [35, 70]. In further studies, an evaluation in a VR setting will be valuable.

10 CONCLUSION

We present QwertyRing, a text entry technique that uses an IMU ring worn on the middle phalanx of the index finger. Users type on any physical surface as if there is a QWERTY keyboard in front of them. They receive text feedback on a separate screen, e.g., an AR/VR headset or an external monitor display. We trained SVM classifiers to recognize touch down and touch up on physical surfaces using an IMU. The accuracies were 99.30% and 99.50%. Based on the sensing of touch down and touch up, QwertyRing can recognize multiple touch gestures such as long presses, swiping and swipe right by using simple rule-based methods. We analyzed the user behavior of typing through a user study and designed the text entry decoder. We evaluated the performance of QwertyRing with the general model and the personal model in a five-day study. For the general model, participants reached a speed of 13.75 WPM in the first 40 minutes and achieved 20.35 WPM at the end. For the personal model, participants started with 13.74 WPM and ended with 20.83 WPM. Results show that QwertyRing is easy to learn to use and outperforms other ring-based techniques. QwertyRing enables efficient touch-based text entry on any desk-like surface, which is valuable in many scenarios such as AR/VR and smart TV.

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China under Grant No. 61521002, a grant from the Institute for Guo Qiang, Tsinghua University, the Natural Science Foundation of China under Grant No. 61672314, and also by Beijing Key Lab of Networked Multimedia. Multimedia.

REFERENCES

- [1] Christoph Amma, Marcus Georgi, and Tanja Schultz. 2012. Airwriting: Hands-free mobile text input by spotting and continuous recognition of 3D-space handwriting with inertial sensors. In *2012 16th International Symposium on Wearable Computers*. IEEE, 52–59.
- [2] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2009. Analysis of text entry performance metrics. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*. IEEE, 100–105.

- [3] Daniel Ashbrook, Patrick Baudisch, and Sean White. 2011. Nenya: subtle and eyes-free mobile input with a magnetically-tracked finger ring. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2043–2046.
- [4] Shiri Azenkot and Shumin Zhai. 2012. Touch behavior with different postures on soft smartphone keyboards. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*. ACM, 251–260.
- [5] Paul Badger. 2018. Capacitive Sensing Library. <https://playground.arduino.cc/Main/CapacitiveSensor>
- [6] Bartosz Bajer, I Scott MacKenzie, and Melanie Baljko. 2012. Huffman base-4 text entry glove (H4 TEG). In *2012 16th International Symposium on Wearable Computers*. IEEE, 41–47.
- [7] Liwei Chan, Yi-Ling Chen, Chi-Hao Hsieh, Rong-Hao Liang, and Bing-Yu Chen. 2015. Cyclopsring: Enabling whole-hand and context-aware interactions through a fisheye ring. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 549–556.
- [8] Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, Chao-Huai Su, Mike Y Chen, Wen-Huang Cheng, and Bing-Yu Chen. 2013. FingerPad: private and subtle interaction using fingertips. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 255–260.
- [9] Barbara S. Chaparro, Jibo He, Colton Turner, and Kirsten Turner. 2015. Is Touch-Based Text Input Practical for a Smartwatch?. In *HCI International 2015 - Posters' Extended Abstracts*, Constantine Stephanidis (Ed.). Springer International Publishing, Cham, 3–8.
- [10] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. 2013. uTrack: 3D input using two magnetic sensors. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 237–244.
- [11] Xiang'Anthony' Chen, Tovi Grossman, and George Fitzmaurice. 2014. Swipeboard: a text entry technique for ultra-small interfaces that supports novice to expert transitions. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 615–620.
- [12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [13] Donald B Devoe. 1967. Alternatives to handprinting in the manual entry of data. *IEEE Transactions on Human Factors in Electronics* 1 (1967), 21–32.
- [14] Leah Findlater, Jacob O Wobbrock, and Daniel Wigdor. 2011. Typing on flat glass: examining ten-finger expert typing patterns on touch surfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 2453–2462.
- [15] Mikael Goldstein and Didier Chincolle. 1999. Finger-joint gesture wearable keypad. In *second workshop on human computer interaction with mobile devices*.
- [16] Jun Gong, Zheer Xu, Qifan Guo, Teddy Seyed, Xiang'Anthony' Chen, Xiaojun Bi, and Xing-Dong Yang. 2018. Wristext: One-handed text entry on smartwatch using wrist gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 181.
- [17] Joshua Goodman, Gina Venolia, Keith Steury, and Chauncey Parker. 2002. Language modeling for soft keyboards. In *Proceedings of the 7th international conference on Intelligent user interfaces*. ACM, 194–195.
- [18] Mitchell Gordon, Tom Ouyang, and Shumin Zhai. 2016. WatchWriter: Tap and gesture typing on a smartwatch miniature keyboard with statistical decoding. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 3817–3821.
- [19] Etienne Grandjean and Karl HE Kroemer. 1997. *Fitting the task to the human: a textbook of occupational ergonomics*. CRC press.
- [20] Tovi Grossman, Xiang Anthony Chen, and George Fitzmaurice. 2015. Typing on glasses: Adapting text entry to smart eyewear. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. ACM, 144–152.
- [21] Yizheng Gu, Chun Yu, Zhipeng Li, Weiqi Li, Shuchang Xu, Xiaoying Wei, and Yuanchun Shi. 2019. Accurate and Low-Latency Sensing of Touch Contact on Any Surface with Finger-Worn IMU Sensor. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. ACM, 1059–1070.
- [22] Jeremy Gummesson, Bodhi Priyantha, and Jie Liu. 2014. An energy harvesting wearable ring platform for gestureinput on surfaces. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*. ACM, 162–175.
- [23] Aakar Gupta and Ravin Balakrishnan. 2016. DualKey: Miniature Screen Text Entry via Finger Identification. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 59–70. <https://doi.org/10.1145/2858036.2858052>
- [24] Aakar Gupta, Cheng Ji, Hui-Shyong Yeo, Aaron Quigley, and Daniel Vogel. 2019. RotoSwipe: Word-Gesture Typing using a Ring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 14.
- [25] Aakar Gupta, Jiushan Yang, and Ravin Balakrishnan. 2018. Asterisk and Obelisk: Motion Codes for Passive Tagging. In *Proceedings of the 31th Annual Symposium on User Interface Software and Technology*. 725–736. <https://doi.org/10.1145/3242587.3242637>
- [26] Chris Harrison and Scott E Hudson. 2009. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. ACM, 121–124.
- [27] Jonggi Hong, Seongkook Heo, Poika Isokoski, and Geehyuk Lee. 2015. SplitBoard: A simple split soft keyboard for wristwatch-sized touch screens. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 1233–1236.
- [28] Nancy Ide. 2008. The american national corpus: Then, now, and tomorrow. In *Selected Proceedings of the 2008 HCSNet Workshop on Designing the Australian National Corpus: Mustering Languages, Summerville, MA. Cascadilla Proceedings Project*.

- [29] Wolf Kienzle and Ken Hinckley. 2014. LightRing: always-available 2D input on any surface. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 157–160.
- [30] Junhyeok Kim, William Delamare, and Pourang Irani. 2018. Thumb-Text: Text Entry for Wearable Devices Using a Miniature Ring. In *Proceedings of Graphics Interface*. 18–25.
- [31] Per-Ola Kristensson and Shumin Zhai. 2004. SHARK 2: a large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*. ACM, 43–52.
- [32] Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto, and Margrit Betke. 2016. EyeSwipe: Dwell-free text entry using gaze paths. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 1952–1956.
- [33] Alan HF Lam, Wen J Li, Yunhui Liu, and Ning Xi. 2002. MIDS: micro input devices system using MEMS sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 2. IEEE, 1184–1189.
- [34] Luis A Leiva, Alireza Sahami, Alejandro Catala, Niels Henze, and Albrecht Schmidt. 2015. Text entry on tiny qwerty soft keyboards. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 669–678.
- [35] Yiqin Lu, Chun Yu, Xin Yi, Yuanchun Shi, and Shengdong Zhao. 2017. Blindtype: Eyes-free text entry on handheld touchpad by leveraging thumb’s muscle memory. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), 18.
- [36] I Scott MacKenzie and R William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI’03 extended abstracts on Human factors in computing systems*. ACM, 754–755.
- [37] Sebastian Madgwick. 2010. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK)* 25 (2010), 113–118.
- [38] Päivi Majaranta and Kari-Jouko Räihä. 2007. Text entry by gaze: Utilizing eye-tracking. *Text entry systems: Mobility, accessibility, universality* (2007), 175–187.
- [39] Anders Markussen, Mikkel R Jakobsen, and Kasper Hornbæk. 2013. Selection-based mid-air text entry on large displays. In *IFIP conference on human-computer interaction*. Springer, 401–418.
- [40] Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. 2014. Vulture: a mid-air word-gesture keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1073–1082.
- [41] Damien Masson, Alix Goguey, Sylvain Malacria, and Géry Casiez. 2017. Whichfingers: Identifying fingers on touch surfaces and keyboards using vibration sensors. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*. ACM, 41–48.
- [42] Anh Nguyen and Amy Banic. 2015. *3DTouch: A wearable 3D input device for 3D applications*. IEEE.
- [43] Tao Ni, Doug Bowman, and Chris North. 2011. AirStroke: bringing unistroke text entry to freehand gesture interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2473–2476.
- [44] Takehiro Niikura, Yoshihiro Watanabe, and Masatoshi Ishikawa. 2014. Anywhere surface touch: utilizing any surface as an input area. In *Proceedings of the 5th Augmented Human International Conference*. ACM, 39.
- [45] Shahriar Nirjon, Jeremy Gummesson, Dan Gelb, and Kyu-Han Kim. 2015. Typingring: A wearable ring platform for text input. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 227–239.
- [46] Masa Ogata, Yuta Sugiura, Hirotaka Osawa, and Michita Imai. 2012. iRing: intelligent ring using infrared reflection. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 131–136.
- [47] Ju Young Oh, Jun Lee, Joong Ho Lee, and Ji Hyung Park. 2017. Anywheretouch: Finger tracking method on arbitrary surface using nailed-mounted imu for mobile hmd. In *International Conference on Human-Computer Interaction*. Springer, 185–191.
- [48] Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. 2013. ZoomBoard: a diminutive qwerty soft keyboard using iterative zooming for ultra-small devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2799–2802.
- [49] Farshid Salemi Parizi, Eric Whitmire, and Shwetak Patel. 2019. AuraRing: Precise Electromagnetic Finger Tracking. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 4, Article Article 150 (Dec. 2019), 28 pages. <https://doi.org/10.1145/3369831>
- [50] Kurt Partridge, Saurav Chatterjee, Vibha Sazawal, Gaetano Borriello, and Roy Want. 2002. TiltType: accelerometer-supported text entry for very small devices. In *Proceedings of the 15th annual ACM symposium on User interface software and technology*. ACM, 201–204.
- [51] Anna Peschock, Julia Duvall, and Lucy E. Dunne. 2014. Argot: a wearable one-handed keyboard glove. In *Acm International Symposium on Wearable Computers: Adjunct Program*.
- [52] Natural Point. 2011. Optitrack. *Natural Point, Inc* (2011).
- [53] Domenico Prattichizzo, Francesco Chinello, Claudio Pacchierotti, and Kouta Minamizawa. 2010. Remotouch: A system for remote touch experience. In *19th International Symposium in Robot and Human Interactive Communication*. IEEE, 676–679.
- [54] Shyam Reyal, Shumin Zhai, and Per Ola Kristensson. 2015. Performance and user experience of touchscreen and gesture keyboards in a lab setting and in the wild. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 679–688.
- [55] Robert Rosenberg and Mel Slater. 1999. The chording glove: a glove-based text input device. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 29, 2 (1999), 186–191.
- [56] Sayan Sarcar, Prateek Panwar, and Tuhin Chakraborty. 2013. EyeK: an efficient dwell-free eye gaze-based text entry system. In *Proceedings of the 11th asia pacific conference on computer human interaction*. ACM, 215–220.

- [57] Alexander Schick, Daniel Morlock, Christoph Amma, Tanja Schultz, and Rainer Stiefelhagen. 2012. Vision-based handwriting recognition for unrestricted text input in mid-air. In *Proceedings of the 14th ACM international conference on Multimodal interaction*. ACM, 217–220.
- [58] R Seibel. 1962. Performance on a five-finger chord keyboard. *Journal of Applied Psychology* 46, 3 (1962), 165.
- [59] Tomoki Shibata, Daniel Afergan, Danielle Kong, Beste F Yuksel, I Scott MacKenzie, and Robert JK Jacob. 2016. DriftBoard: a panning-based text entry technique for ultra-small touchscreens. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM, 575–582.
- [60] Roy Shilkrot, Jochen Huber, Jürgen Steimle, Suranga Nanayakkara, and Pattie Maes. 2015. Digital digits: A comprehensive survey of finger augmentation devices. *ACM Computing Surveys (CSUR)* 48, 2 (2015), 30.
- [61] Massimiliano Solazzi, Antonio Frisoli, and Massimo Bergamasco. 2010. Design of a novel finger haptic interface for contact and orientation display. In *2010 IEEE Haptics Symposium*. IEEE, 129–132.
- [62] Lee Stearns, Uran Oh, Leah Findlater, and Jon E Froehlich. 2018. TouchCam: Realtime Recognition of Location-Specific On-Body Gestures to Support Users with Visual Impairments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 4 (2018), 164.
- [63] KOJI Tsukada and Michiaki Yasumura. 2004. Ubi-finger: A simple gesture input device for mobile and ubiquitous environment. *Journal of Asian Information, Science and Life (AISL)* 2, 2 (2004), 111–120.
- [64] Max G Van Kleek, Wolfe Styke, David Karger, et al. 2011. Finders/keepers: a longitudinal study of people managing information scraps in a micro-note tool. In *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2907–2916.
- [65] Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. VelociTap: Investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 659–668.
- [66] Eric Whitmire, Mohit Jain, Divye Jain, Greg Nelson, Ravi Karkar, Shwetak Patel, and Mayank Goel. 2017. DigiTouch: Reconfigurable Thumb-to-Finger Input and Text Entry on Head-mounted Displays. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 113 (Sept. 2017), 21 pages. <https://doi.org/10.1145/3130978>
- [67] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures (*CHI ’11*). Association for Computing Machinery, New York, NY, USA, 143–146. <https://doi.org/10.1145/1978942.1978963>
- [68] Zheer Xu, Pui Chung Wong, Jun Gong, Te-Yen Wu, Aditya Shekhar Nittala, Xiaojun Bi, Jürgen Steimle, Hongbo Fu, Kening Zhu, and Xing-Dong Yang. 2019. TipText: Eyes-Free Text Entry on a Fingertip Keyboard. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. 883–899.
- [69] Xing-Dong Yang, Tovi Grossman, Daniel Wigdor, and George Fitzmaurice. 2012. Magic finger: always-available input through finger instrumentation. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 147–156.
- [70] Zhican Yang, Chun Yu, Xin Yi, and Yuanchun Shi. 2019. Investigating Gesture Typing for Indirect Touch. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 3, Article 117 (Sept. 2019), 22 pages. <https://doi.org/10.1145/3351275>
- [71] Xin Yi, Chun Yu, Weinan Shi, and Yuanchun Shi. 2017. Is it too small?: Investigating the performances and preferences of users when typing on tiny QWERTY keyboards. *International Journal of Human-Computer Studies* 106 (2017), 44–62.
- [72] Xin Yi, Chun Yu, Weijie Xu, Xiaojun Bi, and Yuanchun Shi. 2017. Compass: Rotational keyboard on non-touch smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 705–715.
- [73] Chun Yu, Yizheng Gu, Zhican Yang, Xin Yi, Hengliang Luo, and Yuanchun Shi. 2017. Tap, dwell or gesture?: Exploring head-based text entry techniques for hmds. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 4479–4488.
- [74] Chun Yu, Ke Sun, Mingyuan Zhong, Xincheng Li, Peijun Zhao, and Yuanchun Shi. 2016. One-dimensional handwriting: Inputting letters and words on smart glasses. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 71–82.
- [75] Cheng Zhang, Anandghan Waghmare, Pranav Kundra, Yiming Pu, Scott Gilliland, Thomas Ploetz, Thad E Starner, Omer T Inan, and Gregory D Abowd. 2017. FingerSound: Recognizing unistroke thumb gestures using a ring. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 3 (2017), 120.
- [76] Cheng Zhang, Xiaoxuan Wang, Anandghan Waghmare, Sumeet Jain, Thomas Ploetz, Omer T Inan, Thad E Starner, and Gregory D Abowd. 2017. FingOrbits: interaction with wearables using synchronized thumb movements. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers*. ACM, 62–65.
- [77] Cheng Zhang, Qiuyue Xue, Anandghan Waghmare, Sumeet Jain, Yiming Pu, Sinan Hersek, Kent Lyons, Kenneth A Cunefare, Omer T Inan, and Gregory D Abowd. 2017. Soundtrak: Continuous 3d tracking of a finger using active acoustics. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1, 2 (2017), 30.
- [78] Cheng Zhang, Qiuyue Xue, Anandghan Waghmare, Ruichen Meng, Sumeet Jain, Yizeng Han, Xinyu Li, Kenneth Cunefare, Thomas Ploetz, Thad Starner, et al. 2018. FingerPing: Recognizing Fine-grained Hand Poses using Active Acoustic On-body Sensing. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 437.
- [79] Mingyuan Zhong, Chun Yu, Qian Wang, Xuhai Xu, and Yuanchun Shi. 2018. Forceboard: Subtle text entry leveraging pressure. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, 528.

- [80] Shengli Zhou, Zhuxin Dong, Wen J Li, and Chung Ping Kwong. 2008. Hand-written character recognition using MEMS motion sensing technology. In *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, 1418–1423.