ThumbText: Text Entry for Wearable Devices Using a Miniature Ring

Junhyeok Kim, William Delamare, Pourang Irani

Graphics Interface Conference 2018, 8–11 May, Toronto, Ontario, Canada

ThumbText: Text Entry for Wearable Devices Using a Miniature Ring

Junhyeok Kim, William Delamare, Pourang Irani
University of Manitoba
Winnipeg, R3T 2N2

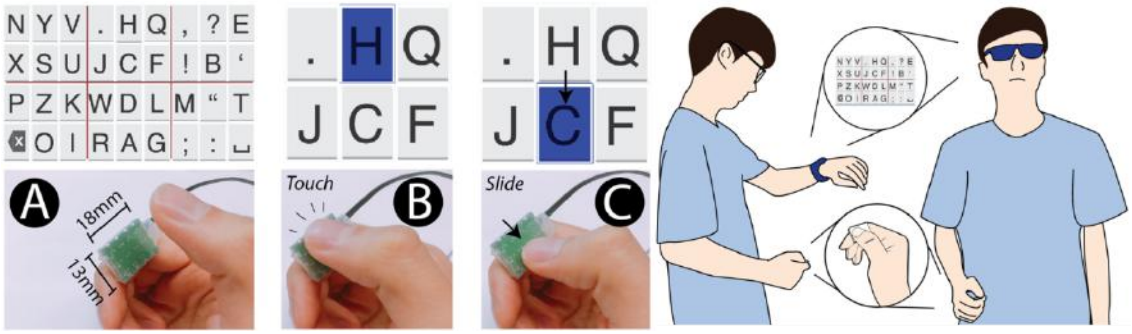

Figure 1: Illustration: typing ‘c’ with ThumbText. (A) The user first locates the character ‘c’ in the top row, middle column (B) then touches the appropriate cell. The second step displays the selected cluster of characters. (C) The user can slide the thumb to reach the ‘c’ character and confirm the selection by lifting the thumb from the screen.

Abstract

Users can benefit from using an auxiliary peripheral that could mitigate many concerns with direct text entry on wearable devices. We introduce ThumbText, a thumb-operated text entry approach for a ring-sized touch surface. Through a multi-part exploration, we first identify a suitable discretization of the miniature touch surface for thumb input. We then design a number of two-step selection techniques for supporting the input of at least 28 characters. On a miniature touch surface, we find that a continuous touch-slide-lift selection technique in a 2×3 grid discretization offers improved performance gains over other selection methods. Finally, we evaluate ThumbText against techniques also designed for wearable devices and find that ThumbText allows for higher text entry rates than SwipeBoard and H4-Writer.

Keywords: Text Entry; Wearables; Design; Smartwatch; Head-Worn Display; Ring; Touch; Thumb.

Index Terms: H.5.2. User Interface. Input devices and strategies.

1 Introduction

Supporting text entry on wearable devices is still an open challenge [12,13,16,24,32,35,44]. This has led researchers to propose novel approaches to address some of the key text entry challenges on emerging wearables, such as smartwatches [7,14,16,32] and head-worn displays (HWDs) [13,38]. However, these approaches are optimized for each device separately and significant design iterations are needed to make them usable across devices [7,13,38]. Ideally one unique text entry mode should be usable across a multitude of wearable devices and thus provide the opportunity to design-and-learn once, and reuse often. Such an approach could mitigate concerns peculiar to any one device, such as finger occlusion [43] and two-handed use on smartwatches, or fatigue [15] and social awkwardness on HWDs [18].

{kimj3415, Pourang.irani} @cs.umanitoba.ca,
{William.Delamare} @acm.org

One promising solution is the use of wearable text entry peripherals [10,12,20,22,25,31,38]. Such devices allow for indirect input and can be used, if designed appropriately, across more than one wearable display. However, existing peripherals come at a cost. Some require users to hold it, which minimizes the use of an entire hand that can often be indispensable in mobile and wearable contexts [21,39]. In other cases, the peripherals require a steep learning curve, making them unusable for short activity bursts [31,41].

For the text entry task on wearable devices, no current peripheral is able to satisfy the following design requirements:

- *miniaturization:* the peripheral should ideally be as small as possible to avoid holding it, but yet not too small to detract significantly from its core task;
- *one-handed operation:* given the requirements for one-handed operation while on-the-go, users should be able to enter text in one-handed mode;
- *self-contained:* for optimum mobility, users should not be required to depend on additional materials or surfaces for text entry; and
- *unified input:* users should be able to apply the same text entry approach across wearable devices;

We propose ThumbText, a peripheral that makes text entry possible via thumb input onto a miniature touchpad affixed to the thumb’s opposing fingers (index or middle finger)**.** As such, ThumbText offers one-handed indirect input, with subtle finger movements [6], and independent of the associated wearable displays. Through a multipart design process, we first assessed the thumb’s dexterity for input on a small touch surface by carefully discretizing the available input space. The ideal discretization patterns were then used to map alphabet characters onto touchpad positions. While operating a smartwatch, we found that ThumbText outperforms SwipeBoard [7] and H4-Writer [27], two very efficient text entry methods applicable to wearable devices.

Our contributions include: (1) ThumbText, a peripheral device and its associated text entry method for wearable devices; (b) a multi-part design process integrating an understanding of the thumb’s dexterity for text entry with ThumbText; and (c) an evaluation demonstrating ThumbText's performance in comparison to existing text entry techniques applicable on a miniaturized touch surface.

2 RELATED WORK

We first present peripherals devised and applicable to text entry in wearable contexts. We then review text entry techniques, making the distinction between one- and multi-step techniques on small input devices.

2.1 Peripheral Device and Applicability

We discuss the relationship between wearable peripheral devices and our design requirements. We only discuss physical devices which come with an associated text entry technique.

Handheld devices: Twiddler [25] is a one-handed keyboard device that allows user to input characters. GesText [20] uses accelerometer based gestures for text entry. Both systems require users to hold a physical device on one hand, preventing users to perform another task while holding it. Handheld devices do not satisfy our *miniaturization* design requirement.

Wristband devices: One-Key Keyboard [22] uses a touch-sensitive surface attached to the wrist for text entry. PalmType [38] uses IR sensors embedded on a wristband to detect touch on the palm. Touch-Sensitive Wristband [10] enables touch interaction directly on the wristband. However, these wristband solutions force users to interact with both hands: one for wearing the wristband, one for typing. Therefore, this does not meet our *one-handed* requirement.

Ring devices: TypingRing [31] allows users to type letters with three fingers on any solid surface. Similarly, TAP [19] allows users to type text with chorded input on nearby surface. Both require a solid surface in the proximity of the user and as such are not *self-contained*.

Wearable rings that provide auxiliary buttons or a touch surface offer a promising solution. They do not need to be held, they can operate in one-handed mode, and are self-contained. Instead of relying on an external surface for input, touch can be directly embedded on the device itself.

2.2 One-Step Text Entry Techniques

One-step text entry techniques on a small device consists in gesture-based input to overcome space limitation issues. EdgeWrite [42] allows users to input letters by performing sequences of hits in the corners of a square, offering improved entry-rates compared to Graffiti [4]. However, input vocabularies [4,42] require users to pre-learn gestures, an often heavy burden for pick-up-and-use contexts such as is common on wearable devices. One solution to overcome memory limitations is to select a letter on a custom keyboard layout. However, techniques enabling users to select 28+ letters on a soft-keyboard need to consider space limitation issues. Two solutions can overcome space limitations: technology-based and interaction-based approaches. Technology-based solutions use statistical inference of imprecise actions. This is the case of WatchWriter [11], InvisiBoard [29], or COMPASS [44]. However, such solutions rely on automatic corrections and predictions that can be detrimental in specific scenarios, e.g., out-of-vocabulary words and non-alphabetical characters [11]. Interaction-based techniques propose to select a letter via an explicit disambiguation step using a second modality (e.g., TiltText [40], TiltType [33] or GesText [20]), such as a wrist rotation. In this work, we focus on techniques using only one modality: one finger touch.

2.3 Multi-Step Text Entry Techniques

We distinguish between techniques focusing on soft keyboard manipulation and techniques focusing on cluster of letters.

First, text entry techniques can propose to manipulate the soft keyboard visualization. With ZoomBoard [32], a first tap area is zoomed in so that a second tap can comfortably select a letter. However, since the area is defined by the first tap, an absolute second tap might not be precise when the input and output spaces are decoupled, as with indirect input. ZShift [23] creates a callout of a magnified portion of the keyboard occluded by the finger touch. However, the zoomed callout requires a display space above the keyboard that might not be always available. Virtual Sliding QWERTY [5] displays only a portion of the soft keyboard to display larger characters. Users can pan the keyboard to reveal hidden characters. On the same principle, SplitBoard [16] displays a half of QWERTY keyboard at a time. Users can swipe horizontally to switch between each half-keyboard. DriftBoard [36] allows users to pan a soft keyboard to select characters via an on-screen pointer at a fixed location.

Second, text entry techniques can propose multiple characters per area followed by a disambiguation step. With Quikwriting [34], once the stylus enters a zone, the user can choose a letter within this zone by moving to another one and moving back to the resting zone. 8Pen [45] builds on Quikwriting and allows users to choose which subgroup they want to select a letter from by moving their finger in a clockwise or counter-clockwise direction. Other techniques adopt the same strategy with different layouts. With MessagEase [30], users perform a first tap to select a cluster of characters, followed by a slide-lift action to further refine the selection. Other techniques build on MessagEase by using different layouts [3], by reducing the number of characters in each area [35], or by integrating a predictive mechanism [9]. The H4 family, e.g., H4-Writer [27] and H4-TEG [2] extend the concept of 'two-steps' to multiple steps depending on the frequency of characters. Thus, frequent characters are selected with two actions, while less frequent characters are selected with 4+ actions. Since H4 operates with only four buttons on a console controller, we hypothesize that this technique can also be adapted on a small ring touchpad. With SwipeBoard [7], users perform two swipe gestures on a smartwatch to enter a letter: the first swipe gesture selects one zone out of nine, the second swipe gesture selects a letter contained in the previously selected zone. SwipeZone [13] extends the SwipeBoard technique to consider the specific form factor of Google Glass' lateral touchpad. SwipeZone extends SwipeBoard since (1) the discrete nature of the input interaction is well suited for limited input spaces, and (2) the technique does not require users to perform precise absolute touch selection. Thus, SwipeBoard is also a good candidate to adapt on a small ring touchpad device.

None of the above techniques were designed for an indirect ring form-factor for text entry on wearable devices. In the remainder of this paper, we present our design process for ThumbText.

3 HARDWARE

We describe our ring prototype, and the hardware and software apparatus used in our studies.

Ring prototype (Figure 1, *A*): The ring consists of a MTCH6102 Capacitive Touch Controller on top of customized printed circuit board. The touch sensitive area is 18mm×13mm with a resolution of 256×160 pixels. The ring device communicates with an Arduino Fio V3 via the I2C protocol.

Apparatus: In the following studies, our main software was implemented in C# with the Unity 5 game engine and ran on a 3.4

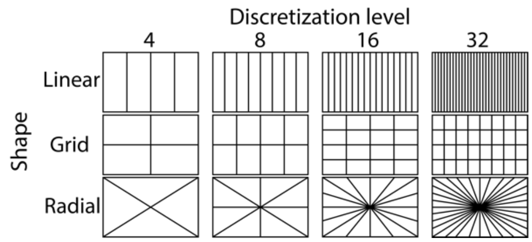

Figure 2: Discretization techniques on the touch surface for identifying the suitable thumb input resolution on our device.

GHz Intel Core i7-3770 with 8GM RAM computer. The ring prototype (i.e., Arduino board) communicated with the main software via USB. Participants were sitting 50cm in front of a Dell U2312HM with 1920x1080 resolution desktop screen. Wearable display devices were an iMacwear M7 smartwatch and the Epson Moverio BT-200 smartglasses. Data communication between the main software and the ring (resp. wearable displays) was done via USB (resp. Wi-Fi).

4 STUDY 1: THUMBTEXT DESIGN

Our final objective is to get users to input 28+ characters. In the first experiment, we explore the performance of thumb input on a ring touchpad. Based on these results, we next explore solutions that can extend the number of selectable targets.

4.1 Experiment 1 – Precision

We look at three experimental factors:

Discretization Shape: We consider the *linear*, *grid,* and *radial* shapes (Figure 2). A *linear* is the simplest form. We chose the horizontal discretization, adapted to the wider form factor of our prototype. A *grid* divides the touchpad into rectangular areas. A *radial* divides the touchpad into triangular areas radiating from the touchpad center.

Discretization Level: We also explored the number of areas that could be defined on the touchpad. We considered *4*, *8*, *16* and *32* areas. Levels of the linear discretization were equal to the number of columns. Levels of the grid discretization lead to 2×2, 2×4, 4×4 and 4×8 grid sizes (Figure 2). We chose more columns than rows to fit our prototype wider form-factor. Levels of the radial discretization were equal to the number of triangles, with all triangles having the same surface area.

*Finger***:** We considered two fingers on which to attach the ring touchpad: the *index* and the *middle* finger. The index finger is the first finger opposite to the thumb: the thumb can hence reach the touchpad quickly. The middle finger allows to keep the index finger free while still allowing a comfortable access to the touchpad.

Participants: 12 participants (5 females), between 19 and 35 years old (M=23.16, SD=4.08), were recruited from local community. 11 of them were right-handed and one was ambidextrous. 11 participants had 3+ years' experience with touch sensitive devices. 5 participants had experience with wearable devices.

Procedure: The study lasted 30 minutes per participants after which they filled a questionnaire for qualitative data. The task consisted in selecting a colored target displayed on a screen. We asked participants not to look at the ring touchpad or their fingers. They were instructed to be as accurate and fast as possible. Participants could practice for a minute before each condition.

Participants were instructed to select a blue target by a tap-and-lift action on the touchpad, with the tap as close as possible to the target. If they needed to correct the landing position, (1) the current

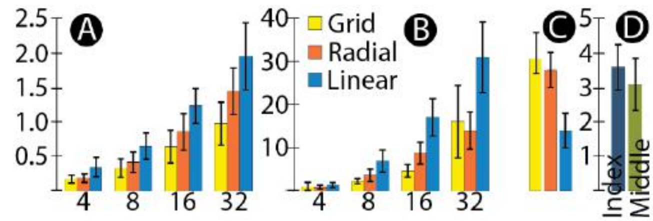

Figure 3: (A) Selection time (s), (B) Error rate (%), and (C) user preference for shapes and (D) for fingers. (95% CI).

area corresponding to the actual thumb position turned red, and (2) participants could slide the thumb toward the target area. Once the thumb was in the target area, the target area turned green and users could lift their thumb to validate the selection. Auditory feedback indicated success or failure, and participants could rest before the next trial.

Experimental Design: We used a repeated-measure within-participant design. The independent variables were the discretization *Shape* (linear, grid, radial), the discretization *Level* (4, 8, 16, 32) and the *Finger* (index, middle). The ordering of *Finger* and *Shape* was counterbalanced across participants using a Latin-square design. The ordering of *Level* followed an increasing difficulty: from level 4 to 32.

The experiment was split into two sections: one for each *finger*. Each section contained three blocks, i.e., one block per *shape*. Each block consisted in 4 series of trials, one per discretization *level*. For each condition, we collected 20 successful selections. In case of an error, the selection was re-queued to the pool of remaining selections. This design ensured the collection of 3 *shapes* × 4 discretization *levels* × 2 *fingers* × 20 successful selections = 480 acquisitions per participants, hence a total of 5760 acquisitions.

4.2 Experiment 1 – Results

The main dependent measures for the task were the selection time and the error rate defined as: *ER*= $\frac{\#Errors}{\#Success + \#Errors}$. We collected a total 8340 trials, including 5760 successful selections and 2580 failed selections.

Error rate: We found a significant effect of *Shape* on the error rate [$F_{2,26}$=49.81, p<0.001, η_p^2=0.82]: Grid offers a better accuracy than radial [p<0.01], which gives a better accuracy than linear [p<0.001]. We also found a significant effect of *Level* [$F_{3,26}$=70.23, p<0.0001, η_p^2=0.86] with each *level* being significantly more error prone than the previous smaller one [p<0.001]. We did not find any significant effect of *Finger* on the error rate [$F_{1,26}$=4.32, p=0.06]. The grid and radial *shapes* are significantly different only for the discretization *level* 16 [$F_{2,33}$=22.04, p<0.001, η_p^2=0.67] (Figure 3, B). In contrast, the linear *shape* is significantly different from the two other *shapes* as soon as we reach *level* 8 [$F_{2,33}$=15.85, p<0.001, η_p^2=0.59].

Selection time: We consider only successful trials for the selection time analysis. We applied a log transform to our data to satisfy the normality and homogeneity of variances assumptions. Figures show non-transformed data. We performed two-ways ANOVAs and accounted for repeated measures by treating participants as a random variable. We used multiple pairwise t-test comparisons with a Bonferroni correction for post-hoc tests. We found a significant main effect of *Shape* [F2,26=48.49, p<0.00001, ηp2=0.82] and *Levels* [F3,26=96.97, p<0.00001, ηp2=0.90], but no effect of *Finger* [F1,26=0.75, p=0.40] on the selection time. Post-hoc tests reveal a significant difference between all *shapes* [p<0.001]. Grid leads to faster selection times than radial, which

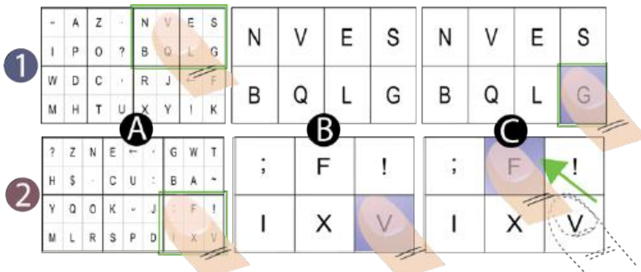

Figure 4: (1) Illustration of Tap-4-8. (2) Lift-6-6. A) User touches an area. B) Second step displaying the characters included in the previously selected area. C) Users (1) tap or (2) slide toward the final character to select it.

leads to faster selection times than linear. We also found significant differences between all discretization *levels* [$p<0.001$]: the more discretized areas, the longer the selection time (Figure 3, A). We did not find any interaction effect.

4.3 Experiment 1 – Discussion

From the results, we infer the need for a two-step interaction technique to input 28+ characters on our ring touchpad. Large discretization *levels* led to poor performances in terms of both the selection time and the accuracy. Since (1) the grid-shaped discretization offers a better precision than the linear and radial discretization, and (2) participants' preferences lean toward the grid discretization (Figure 3, C), we focus on two-step interaction techniques using the grid-shaped discretization. Regarding the *finger* to which the ring touchpad is attached, we did not find any significant difference. However, it seems that participants preferred using the ring on the index finger (Figure 3, D). We hence use the index finger for the remaining studies.

4.4 Experiment 2: Multi-Steps Selection

Our approach to increase the number of selectable areas on a ring touchpad involves a two-step concept [7,30,35]: a top-level for selecting a cluster of letters, and a lower-level for selecting a character.

- *Grid4-Grid8*: We first consider the simplest grid shaped discretization: 4 areas. The second step can use a discretization of 8 areas, offering 4×8 = 32 characters (Figure 4, 1). Once a top-level area is selected, the associated characters are displayed in a grid-8 design. The second step has two potential drawbacks. First, a discretization level of 8 is more difficult to use than a smaller one. Second, the two steps use a different layout which could impose a cognitive load on users when transitioning between these two steps.
- *Grid6-Grid6*: We hence consider another alternative. A discretization level of 6 is in-between the 4 and 8 levels. It offers a good compromise for both the first and second steps. In addition, since both steps use the same layout, users do not have to switch their mental model of the discretization during the transition. This design offers the possibility to select 6×6 = 36 characters (Figure 4, 2).

For both designs, we explore different input techniques:

- *Tap*: users perform (i) a first tap to select a cluster of characters from the top-level cells, and (ii) a second tap to select a character from the lower-level cells (Figure 4, 1). We designate designs using the 'tap' keyword and the discretization used, i.e., Tap-4-8 and Tap-6-6.
- *Touch and lift*: users (i) touch an area to select a cluster of characters, (ii) can slide their thumb to navigate in the lower-level, and (iii) validate their selection by lifting their thumb from the touchpad (Figure 4, 2). This input technique allows users to select a 'main character' in each top-level area without sliding to other discretized lower-level area, i.e., 4 with Grid4-Grid8 and 6 with Grid6-Grid6. Such main characters could be based on the letter frequency in the user's language. We designate these designs using the 'lift' keyword and the discretization used, i.e., Lift-4-8 and Lift-6-6.

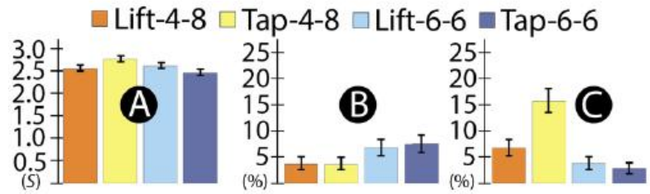

Figure 5: Experiment 2 result (A) Selection time (s) (B) Soft error (%), and (C) Hard error (%)

Participants: 8 new participants (4 females), ages between 19 and 37 (M=27.1, SD=7.98) volunteered for the experiment. All of them had experience with touch sensitive devices, three of participants had < 1-year experience with wearable devices.

Procedure: The experiment lasted about 1 hour per participants after which they filled a questionnaire to get qualitative data. Participants selected a character using one of the four designs described above. A trial consisted in a character selection. The character to select was displayed on top of layout. Visual feedback consisted in a blue coloration of the area in which the thumb was on. Auditory feedback indicated the success of the selection. Participants could rest before starting the next trial. Minimal training was allowed to ensure that participants understood the task.

Experimental Design: The task used a repeated-measure within-participant design with *Designs* (Tap-4-8, Tap-6-6, Lift-4-8, Lift-6-6) as an independent variable. The ordering of *Designs* was counterbalanced across participants using a Latin-square design.

The task was divided into 4 blocks: one for each *Design*. Each block consisted of 10 sequences of trials. In each sequence, participants had to select 8 letters. The 8 letters and character positions were randomly chosen for the 1st sequence and remained the same until switching the *Design*. We then considered two distinct repetition blocks with 5 successive sequences of trials each. The independent variables were the *Design* and the *Repetition* (1, 2). We collected 4 designs × 8 characters × 10 repetitions = 320 acquisitions per participant, for a total of 2560 acquisitions.

4.5 Experiment 2 – Results

The main dependent measures for the task were the selection time and the error rate. We also distinguished between *soft errors* and *hard errors* [7], i.e., errors on the first step (selection of a cluster of characters) and the second step (the actual character selection). We removed 1.46% of the data considered as outliers, i.e., with a selection time more than three standard deviations from the mean.

Error rate: We found a significant main effect of *Design* [$F_{3,56}=6.71$, $p<0.01$, $\eta_p^2=0.49$] on error rate. Tap-4-8 leads to more errors than all the other designs [$p<0.05$]. We did not find any improvement over time [$F_{1,56}=0.00001$, $p=0.99$]. There was no significant difference between *Design* regarding soft errors [$F_{3,56}=2.64$, $p=0.08$]. For the hard error type, we found a significant main effect of *Design* [$F_{3,56}=14.67$, $p<0.0001$, $\eta_p^2=0.68$]. Not surprisingly, Tap-4-8 leads to more hard errors than all the other designs [$p<0.01$] (Figure 4).

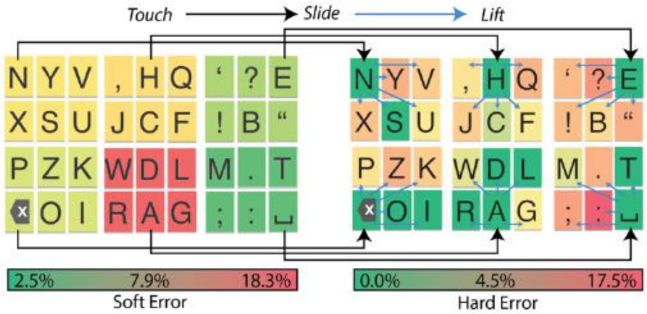

Figure 6: Heatmap for Lift 6-6 based on the experiment 2 results.

Selection time: We performed two-way ANOVAs on log-transformed successful data and accounted for repeated measures by treating participants as a random variable. We used multiple pairwise t-test comparisons with a Bonferroni correction for post-hoc tests. We did not find any significant effect of *Design* on the selection time [$F_{3,56}$=2.23, p=0.12]. There was a learning effect [$F_{1,56}$=88.41, p<0.0001, η_p^2=0.93]: participants were faster at the end than at the beginning with all designs. There was no interaction effect.

4.6 Experiment 2 – Discussion

Except for Tap-4-8, all designs hold promising results, without significant differences regarding the selection time and the error rate. Participants preferred the 6-6 design (Figure 4, 2). However, there is no clear distinction between Tap and Touch-and-Lift input methods. Yet, "*[with Tap] there is no going-back in case of error*" (P1). Thus, we chose Lift-6-6 as a basis for ThumbText.

We also generated a heatmap (Figure 6) based on 'lift-6-6' results. As landing precision (soft error) indicates, the bottom-center area (where 'W','D','L','R','A', and 'G' are) was the most error-prone cell. However, once participants managed to land correctly in this cell, the following selection of characters was easy (hard error).

4.7 Resulting ThumbText Design

ThumbText uses a 2×3 discretization of the ring touchpad for both steps of the character selection: selecting a cluster (tap) and selecting a character (slide and lift). ThumbText hence extends the family of techniques proposing multiple characters per area [34,45], disambiguated by a slide-and-lift action [30,35]. For the ThumbText final design, we also optimized our layout based on the frequency of English letters: most frequent letters are placed in less error-prone cells (Figure 6). Although a near-QWERTY layout could be applied, we chose to consider the precision difference between cells at the extra cognitive cost for novice users to learn this new layout.

5 STUDY 2: THUMBTEXT EVALUATION

We evaluate ThumbText against existing baseline techniques in a context deliberately difficult for ThumbText (Experiment 3), and in a normal context (Experiment 4).

5.1 Experiment 3: ThumbText Evaluation – Difficult Context

We explore how ThumbText performs under difficult conditions against existing techniques, namely SwipeBoard [7], and H4-Writer [27]. Our baseline techniques demonstrate considerably high performance in their original evaluations (19.58 words-per-minute, 20.4 WPM, respectively). H4-Writer's layout uses only four areas, which should lead to good results on our ring touchpad based on results from experiment 1 (Figure 3). SwipeBoard uses swipe gestures to select characters. We hence want to evaluate how Tap (H4-Writer), and Touch-and-lift (ThumbText) perform against swipe gestures that is not considered in the previous study.

The study uses our ring prototype as the input device for all three techniques, and the iMacwear M7 smartwatch as a wearable display. All techniques are hence used as an indirect text entry method. The layout was divided into a 9-cells to replicate SwipeBoard, and into a 4-cells radial layout for H4-Writer.

We adapted the protocol from Chen at al. to simulate expert performances [7]. We used a subset of 6 characters deliberately difficult for ThumbText, instead of choosing letters fair for all techniques – which might be impossible without introducing a bias. This also allows us to determine the lowest performance of our technique: any other context will hence lead to similar or better text entry rates. We chose the letters ('U', 'F', 'B', 'P', 'I', 'L'), leading to 17 4-letters words such as 'FLIP' or 'BLIP'. With ThumbText, these letters (i) all require a sequence of touch-slide-lift actions, and (ii) are placed on error-prone cells (Figure 6). For H4-Writer, this resulted in 1 character using 2 taps, 4 characters using 3 taps, and 1 character using 4 taps. With SwipeBoard, all characters require two actions.

Experimental Design: The task used a repeated-measure within-participant design with *Technique* (ThumbText, SwipeBoard, H4-Writer) and *Block* (1 to 8) as independent variables. The ordering of *Technique* was counterbalanced across participants using a Latin-square design.

The experiment consisted of two sessions separated by at least 2h and at most by 24h. A session was divided into 3 sections: one per *Technique*. Each section consisted of 4 blocks, with each block involving 5 sets. A set consisted of 7 words, each randomly picked in the list of 17 words generated via our 6 letters. We collected 2 sessions × 3 sections × 4 blocks × 5 sets × 7 words = 840 words per participants, for a total of 7560 acquisitions.

Participants: We recruited 9 new participants (3 females), aged between 20 and 25 (M=22, SD=1.41). Participants received a $15 gift card for their participation. All participants had more than 3 years of touch sensitive devices.

Procedure: The experiment lasted about 90 minutes per session and per participant. Participants filled a questionnaire to get qualitative data after the second session. Participants could rest between sets. The target word was displayed on top of the smartwatch screen, with the transcribed word right underneath. In case of an incorrect input, the transcribed word turned red and participants had to correct the error. A trial automatically ended as soon as the transcribed word matched the target word. No training was provided as we wanted to evaluate the learning process.

5.2 Experiment 3: Results

We report standard text entry metrics when applicable [1,37]. For instance, since participants had to input the correct word to end a trial, the number of Unnoticed Error and the Minimum String Distance are always 0, leading to a Minimum String Distance Error Rate of 0%. We focus our analysis on three quantitative metrics (words-per-minute, total error rate, and keystroke per character [1]) and on qualitative feedback. We performed two-way ANOVAs on log-transformed data and accounted for repeated measures by treating participants as a random variable. Post-hoc tests used multiple pairwise t-test comparisons with a Bonferroni correction. Figures show non-transformed data.

Figure 7: Experiment 3 – Difficult result.

Each metric is calculated with the following formulas [1]:

$$WPM = \frac{|Transcribed| - 1}{Seconds} \times 60 \times \frac{1}{5}$$

$$Total\ ER = \frac{Unnoticed\ Er + Corrected\ Er}{|Corrected| + Unnoticed\ Er + Corrected\ Er} \times 100\%$$

$$KSPC = \frac{|Input\ Stream|}{|Transcribed|}$$

Words-Per-Minute (WPM): We found a significant main effect of *Technique* [$F_{2,18}$=14.64, p<0.0001, η_p^2=0.65] and *Block* [$F_{7,18}$=35.06, p<0.0001, η_p^2=0.81] on the text entry rate. Post-hoc tests show that ThumbText (8.47±2.45 wpm) allows for higher wpm values than SwipeBoard (6.96±1.29 wpm) [p<0.05] and H4-Writer (6.19±2.02 wpm) [p<0.05] in the 8th block. We didn't find interaction effect between *Technique* and *Block*.

Total Error Rate (Total ER): We did not find any significant effect from *Technique* on the total error rate [$F_{2,18}$=2.27, p=0.13]. We found a significant main effect of *Block* on the total error rate [$F_{7,18}$=3.47, p<0.005, η_p^2=0.30], as participants managed to decrease the number of errors from 11.96±7.08% in the 1st block to 7.94±3.96% in the last block. We also found an interaction effect between *Technique* and *Block* [$F_{14,18}$=2.25, p<0.05] (Figure 7).

Keystrokes Per Character (KSPC): We found a significant main effect of *Technique* [$F_{2,18}$=25.77, p<0.0001, η_p^2=0.76] and *Block* [$F_{7,18}$=6.30, p<0.0001, η_p^2=0.44] on the KSPC. We also found an interaction effect between *Technique* and *Block* [$F_{14,18}$=2.94, p<0.005, η_p^2=0.26]. With SwipeBoard, KSPC reaches a plateau after the 3rd block around 2.75±0.61. Nearly constant KSPC values indicates that the increase of text entry rates is most likely due to a motor-skills learning. With ThumbText, KSPC reaches a plateau after 2nd block around 3.53±0.29.

5.3 Experiment 4: ThumbText Evaluation – Normal Context

We next evaluate ThumbText without focusing on characters explicitly difficult to input. For this, we chose characters based on an optimized H4-Writer. H4-Writer optimizes the complete set of characters based on their frequency. Evaluating techniques with characters spread out along this frequency range allows us to evaluate techniques in a normal context (i.e., not designed to be difficult or easy).

In this comparison, we chose letters spread out across the range of number of Tap action of H4-Writer: ('E', 'A', 'D', 'R', 'B', 'Z'), leading to a set of 14 4-letters words (e.g., 'BADE' and 'RAZE'). This allows us to evaluate technique based on the English letter frequency. Thus, H4-Writer uses two characters using 2 taps, two characters using 3 taps, one character using 4 taps, and one character using 5 taps. ThumbText uses two characters needing 1 tap, and four characters needing a sequence of tap-slide-lift actions. With SwipeBoard, all characters require two actions. We expect H4-Writer's performance to incur a drop of performance compared to the previous study, and SwipeBoard's performance to remain the same.

Figure 8: Experiment 4 - Normal result.

Participants: We recruited 9 new participants (3 females), aged between 22 and 30 (M=26.1, SD=2.84) for the experiment. All of them had more than 2 years of experience with touch sensitive devices and one participant had more than 3 years of experience with wearable devices (smart wristband). The experiment followed the exact same protocol and procedure as Experiment 3.

5.4 Experiment 4 – Results

Overall, text entry metrics (i.e., WPM, Total ER, and KSPC) follow the same trends as in the previous experiment.

WPM: We found a significant main effect of *Technique* [$F_{2,18}$=35.96, p<0.0001, η_p^2=0.82] and *Block* [$F_{7,18}$=130.74, p<0.0001, η_p^2=0.94] on the text entry rate. Post-hoc tests show that ThumbText (11.41±2.30 wpm in 8th block) allows for higher wpm values than SwipeBoard (6.49±1.26 wpm in 8th block) [p<0.0001] and H4-Writer (6.83±1.88 wpm 8th block) [p<0.0001]. We also found an interaction effect [$F_{14,18}$=1.80, p<0.05, η_p^2=0.18]: all techniques indicate a learning effect. However, the learning reaches a plateau after the 4th block with H4-Writer, and the 6th block with SwipeBoard and ThumbText. Interestingly, SwipeBoard is significantly faster than H4-Writer only during the first block [p<0.05], while ThumbText provides faster entry-rates than the two other techniques in all blocks [p<0.01] (Figure 8).

Total ER: As in the previous experiment, we did not find any significant effect from *Technique* on the total error rate [$F_{2,18}$=3.51, p=0.055]. We found a significant main effect of *Block* on the total error rate [$F_{7,18}$=2.91, p<0.05, η_p^2=0.27], as participants managed to decrease the number of errors from 13.30±7.21% in the 1st block to 9.08±3.27% in the last block. We did not find any interaction effect [$F_{14,18}$=1.21, p=0.27]: only SwipeBoard demonstrates a higher error rate than the other techniques during the first block, but leads to approximately the same values on the second block (Figure 8).

KSPC: When using ThumbText and H4-Writer, participants increased their text entry rate without significantly decreasing their error rate. Constant KSPC values support this observation (p>0.05) (Figure 8). Thus, the increase of text entry rates is most likely due to a motor-skills learning. On the contrary, with SwipeBoard, participants decreased their error rate while also reducing the KSPC value (p<0.05). This indicates that participants learnt how to use SwipeBoard by mostly reducing the number of input errors, and hence their input actions.

5.5 Experiment 3 and 4 – Discussion

WPM: Both cases, 8.46±2.45 wpm (experiment 3 – difficult context) and 11.41±2.30 wpm (experiment 4 – normal context) show that our technique, ThumbText, perform significantly better than the other techniques. For SwipeBoard, no significant difference was observed as predicted (6.96±1.29 wpm, 6.49±1.26 wpm). With H4-Writer, however, text entry rates improved from 6.19±2.02 wpm to 6.83±1.88 wpm as predicted.

Total ER: From the results, we showed that no significant differences between two experiments were observed as in the first block (11.96±7.09%, 13.30±7.21%) and the last block (7.95±3.96%, 9.08±3.27%). These similar differences between the first block and the last block (approximately 4% each) validate our initial hypothesis. Note that with more participants, some significant differences with little effect size could appear.

KSPC: ThumbText dropped its KSPC values from experiment 3 (3.86 in the first block, 3.39 in the last block) to experiment 4 (2.70, 2.53 respectively) as predicted due to different character set used. SwipeBoard and H4-Writer, however, didn't change their KSPC values as predicted.

First, we showed that ThumbText allows for faster text entry rates than SwipeBoard and H4-Writer. We argue that ThumbText, designed step-by-step, is hence more suitable than state-of-the-art techniques adapted to a ring touchpad.

Second, we note that in this work, SwipeBoard (6.96, 6.49 wpm) and H4-Writer (6.19, 6.83 wpm) do not reach performances reported in their original work, 19.58 wpm and 20.4 wpm respectively. One hypothesis for this drop of performance is the different input devices used in these studies, as already hypothesized in previous work [36].

Finally, we asked participants' feedbacks on each technique they used in experiments. Qualitative results show that participants reported that SwipeBoard's QWERTY layout was an asset compared to the other techniques (P3, P4, P7, P8). Also, the idea is intuitive (P3, P4, P7, P9, P2') and the cancel action is a real advantage (P5, P9). H4-Writer was easy to use (P1, P9, P1', P2'), but participants noted the learning curve (P1, P2, P3, P5, P6, P7) and the number of taps to enter a character (P3, P4, P7). Participants also noticed the unusual layout used by ThumbText (P2, P6, P7, P8). However, participants considered ThumbText as fast, convenient (P1, P3, P7, P9, P1', P2'), and easy to use (P1, P5, P8). Overall, participants preferred ThumbText and H4-Writer to SwipeBoard.

6 LIMITATIONS

We did not integrate any correction or prediction mechanism in our study as we were essentially focusing on human performances. Such mechanism could create artificial problems (e.g., wrong auto-correction) and/or artificial advantages (e.g., good prediction). These artificial effects would have prevented us from understanding how the ring touchpad could effectively be used for text entry purposes. Although we were able to determine the users-only performances (8.47 wpm in a difficult context), we did not evaluate the full potential of ThumbText offered by standard text entry technique enhancements. Results from previous work show a maximum of 82% increase in text entry rates depending on the presented words [17,28]. This indicates that ThumbText could theoretically allow for well over 20 wpm. However, this result remains to be scientifically validated via user experiments.

We identified a few limitations on our analysis and experimental design. The difficult word set for experiment 3 was based on our resulting heatmap, and normal words were chosen based on their English letter frequency. Thus, experiment 3 covers 20% of English characters' frequency, and experiment 4 covers 32%. While popular phrase sets cover 95% of letters frequency [26], we covered over 50% of letter frequency with participants considered at expert-level [7]. In these experiments, we showed that ThumbText outperformed two existing techniques, with both infrequent (experiment 3) and frequent letters (experiment 4). Beside an evaluation with experts and full sentences, ThumbText remains to be adapted for full text edition tasks (e.g., editing font size, font decoration, font colors, etc.) to be considered a complete text-editing input technique. Note that the sample sizes were not large in experiment 3 and 4. However, our effect sizes, which are independent from sample sizes, were unanimously high according to Cohen's rule of thumb [8]. We also observed *p values* <.05. Therefore, we believe that our data, even with small sample sizes, captured phenomena nicely.

7 CONCLUSION AND FUTURE WORK

The growing use of wearable devices opens opportunities for novel input techniques. This requires users to adapt to the way they interact with each device. We propose ThumbText, a text entry technique that works across multiple wearable devices to help users from learning multiple device-dependent text entry techniques.

With ThumbText, users select characters on a ring touchpad. This setup allows ThumbText to meet multiple requirements for wearable text entry situations such as one-handed input. We report the results of a multipart design process. ThumbText uses a two-step selection process with a seamlessly continuous touch-slide-lift action of the thumb on the touchpad. The ring touchpad is discretized using a 2×3 grid. Each cell contains 6 characters. First, users select a group of letters by touching the ring touchpad. The 6 corresponding characters fill the 6 cells. For the second step, users can slide their thumb on the touchpad and lift to confirm the selection. We designed ThumbText so that frequent letters are positioned on less error-prone cells. We validated ThumbText and demonstrated that (1) ThumbText offers faster text entry speed than established wearable techniques, namely SwipeBoard and H4-Writer, and (2) ThumbText can be used across multiple wearable displays without loss of performance.

In addition to the evaluation of ThumbText with auto-correct and prediction features, we plan to extend this work by making the most of our prototype in *ecological longitudinal* studies. Our working prototype can be connected to smartwatches and smartglasses via Bluetooth. We will be able to assess expert performance in the wild and compare this to our in-lab study. Furthermore, our early prototype, which we restricted to one phalange can be designed more ergonomically to fit onto the user's finger. Methods for precise touch detection over the user's skin will also be experimented with to design more seamless text input approaches for wearable devices.

ACKNOWLEDGMENTS

We thank the NSERC CRC program for supporting this work.

REFERENCES

[1] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2009. Analysis of text entry performance metrics. In *Proc. TIC-STH*, 100–105.

[2] Bartosz Bajer, I. Scott MacKenzie, and Melanie Baljko. 2012. Huffman Base-4 Text Entry Glove (H4 TEG). In *Proc. ISWC*, 41–47.

[3] Nikola Banovic, Koji Yatani, and Khai N. Truong. 2013. Escape-Keyboard: A Sight-Free One-Handed Text Entry Method for Mobile Touch-screen Devices. *International Journal of Mobile Human Computer Interaction* 5, July: 42–61.

[4] Steven J. Castellucci and I. Scott MacKenzie. 2008. Graffiti vs. unistrokes: an empirical comparison. In *Proc. CHI*, 305–308.

[5] Jae-Min Cha, Eunjung Choi, and Jihyoun Lim. 2015. Virtual Sliding QWERTY: A new text entry method for smartwatches using Tap-N-Drag. *Applied Ergonomics* 51, June: 263–272.

[6] 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 *Proc. UIST*, 255–260.
[7] 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 *Proc. UIST*, 615–620.
[8] J Cohen. 1988. Statistical power analysis for the behavioral sciences. *Statistical Power Analysis for the Behavioral Sciences 2nd*, 567.
[9] FlickKey. 2010. FlickKey Keyboard. Retrieved from http://www.flickkey.com
[10] Markus Funk, Alireza Sahami, Niels Henze, and Albrecht Schmidt. 2014. Using a touch-sensitive wristband for text entry on smart watches. *Proc. CHI EA*: 2305–2310.
[11] Mitchell Gordon, Tom Ouyang, and Shumin Zhai. 2016. WatchWriter: Tap and Gesture Typing on a Smartwatch Miniature Keyboard with Statistical Decoding. In *Proc. CHI*, 3817–3821.
[12] Timo Götzelmann and Pere-Pau Vázquez. 2015. InclineType – An Accelerometer-based Typing Approach for Smartwatches. In *Proc. Interacción*, 1–4.
[13] Tovi Grossman, Xiang Anthony Chen, and George Fitzmaurice. 2015. Typing on Glasses: Adapting Text Entry to Smart Eyewear. In *Proc. MobileHCI*, 144–152.
[14] Aakar Gupta and Ravin Balakrishnan. 2016. DualKey: Miniature Screen Text Entry via Finger Identification. In *Proc. CHI*, 59–70.
[15] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed Endurance: A metric to quantify arm fatigue of mid-air interactions. In *Proc. CHI*, 1063–1072.
[16] Jonggi Hong, Seongkook Heo, Poika Isokoski, and Geehyuk Lee. 2015. SplitBoard: A Simple Split Soft Keyboard for Wristwatch-sized Touch Screens. In *Proc. CHI*, 1233–1236.
[17] Heidi M. Horstmann and Simon P. Levine. 1991. The effectiveness of word prediction. In *Proc. RESNA*, 100–102.
[18] Yi-Ta Hsieh, Antti Jylhä, Valeria Orso, Luciano Gamberini, and Giulio Jacucci. 2016. Designing a Willing-to-Use-in-Public Hand Gestural Interaction Technique for Smart Glasses. In *Proc. CHI*, 4203–4215.
[19] Tap Systems Inc. TAP. Retrieved from http://www.tapwithus.com
[20] Eleanor Jones, Jason Alexander, Andreas Andreou, Pourang Irani, and Sriram Subramanian. 2010. GesText: Accelerometer-Based Gestural Text Entry Systems. In *Proc. CHI*, 2173.
[21] Amy K. Karlson and Benjamin B. Bederson. 2008. One-handed Touchscreen Input for Legacy Applications. *Proc. CHI*: 1399–1408.
[22] Seoktae Kim, Minjung Sohn, Jinhee Pak, and Woohun Lee. 2006. One-key Keyboard: A Very Small QWERTY Keyboard Supporting Text e Entry for Wearable Computing. In *Proc. OZCHI*, 305–308.
[23] Luis a Leiva, Alireza Sahami, Alejandro Catala, Niels Henze, and Albrecht Schmidt. 2015. Text Entry on Tiny QWERTY Soft Keyboards. In *Proc. CHI*, 669–678.
[24] Yiqin Lu, Chun Yu, X I N Yi, Yuanchun Shi, and Computer Science. 2017. BlindType : Eyes-Free Text Entry on Handheld Touchpad by Leveraging Thumb ' s Muscle Memory. In *Proc. IMWUT* 1, 2: June: 1–24.
[25] Kent Lyons, Thad Starner, Daniel Plaisted, James Fusia, Amanda Lyons, Aaron Drew, and E W Looney. 2004. Twiddler typing: one-handed chording text entry for mobile phones. *Proc. CHI*, 671–678.
[26] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *Proc. CHI*, 754–755.
[27] I. Scott MacKenzie, R William Soukoreff, and Joanna Helga. 2011. 1 thumb, 4 buttons, 20 words per minute: design and evaluation of H4-writer. In *Proc. UIST*, 471–480.
[28] Tina Magnuson and Sheri Hunnicutt. 2002. Measuring the effectiveness of word prediction: The advantage of long-term use. *TMH-QPSR* 43, 1: 57–67.
[29] Aske Mottelson, Christoffer Larsen, Mikkel Lyderik, Paul Strohmeier, and Jarrod Knibbe. 2016. Invisiboard: maximizing display and input space with a full screen text entry method for smartwatches. In *Proc. MobileHCI*, 53–59.
[30] Saied B. Nesbat. 2003. A system for fast, full-text entry for small electronic devices. In *Proc. ICMI*, 4–11.
[31] Shahriar Nirjon, Jeremy Gummeson, Dan Gelb, and Kyu-han Kim. 2015. TypingRing: AWearable Ring Platform for Text Input. In *Proc. MobiSys*, 227–239.
[32] Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. 2013. ZoomBoard: a diminutive qwerty soft keyboard using iterative zooming for ultra-small devices. In *Proc. CHI*, 2799–2802.
[33] Kurt Partridge, Saurav Chatterjee, Vibha Sazawal, Gaetano Borriello, and Roy Want. 2002. TiltType: Accelerometer-Supported Text Entry for Very Small Devices. In *Proc. UIST*, 201–204.
[34] Ken Perlin. 1998. Quikwriting: Continuous Stylus-based Text Entry. In *Proc. UIST*, 215–216.
[35] Yuan-Fu Shao, Masatoshi Chang-Ogimoto, Reinhard Pointner, Yu-chih Lin, Chen-ting Wu, and Mike Chen. 2016. SwipeKey: a swipe-based keyboard design for smartwatches. In *Proc. MobileHCI*, 60–71.
[36] Tomoki Shibata, Daniel Afergan, Danielle Kong, Beste F Yuksel, I. Scott MacKenzie, and Robert J.K. Jacob. 2016. DriftBoard: A Panning-Based Text Entry Technique for Ultra-Small Touchscreens. In *Proc. UIST*, 575–582.
[37] R William Soukoreff and I. Scott MacKenzie. 2003. Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric. In *Proc. CHI*, 113–120.
[38] Cheng-Yao Wang, Wei-Chen Chu, Po-Tsung Chiu, Min-Chieh Hsiu, Yih-Harn Chiang, and Mike Y Chen. 2015. PalmType: Using Palms as Keyboards for Smart Glasses. In *Proc. MobileHCI*, 153–160.
[39] Yuntao Wang, Chun Yu, Jie Liu, and Yuanchun Shi. 2013. Understanding performance of eyes-free, absolute position control on touchable mobile phones. In *Proc. MobileHCI*, 79–88.
[40] Daniel Wigdor and Ravin Balakrishnan. 2003. TiltText. In *Proc. UIST*, 81–90.
[41] Jacob O. Wobbrock. 2006. The Future of Mobile Device Research in HCI. In *CHI 2006 Workshop*, 131–134.
[42] Jacob O. Wobbrock, Brad A. Myers, and John A. Kembel. 2003. EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. In *Proc. UIST*, 61–70.
[43] Haijun Xia, Tovi Grossman, and George Fitzmaurice. 2015. NanoStylus: Enhancing Input on Ultra-Small Displays with a Finger-Mounted Stylus. In *Proc. UIST*, 447–456.
[44] Xin Yi, Chun Yu, Weijie Xu, Xiaojun Bi, and Yuanchun Shi. 2017. COMPASS: Rotational Keyboard on Non-Touch Smartwatches. In *Proc. CHI*, 705–715.
[45] 8Pen. Retrieved from http://www.8pen.com/