TEXT-ENTRY ON A MINIATURE PERIPHERAL FOR WEARABLE DEVICES

JUNHYEOK KIM

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfilment of the requirements of the degree of

MASTER OF SCIENCE

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada

August 2018

ABSTRACT

Users can benefit from using an auxiliary peripheral that could mitigate many concerns with direct text-entry on wearable devices. I introduce ThumbText, a thumb-operated text-entry approach for a ring-sized touch surface. Through a multi-part exploration, I first identify a suitable discretization of the miniature touch surface for thumb input. I then design a number of two-step selection techniques for supporting the input of at least 28 characters. On a miniature touch surface, I find that a continuous touch-slide-lift selection technique in a 2×3 grid discretization offers improved performance gains over other selection methods. Finally, I evaluate ThumbText against techniques also designed for wearable devices and find that ThumbText (11.41 words-per-minute) allows for higher text-entry rates than SwipeBoard (6.49 words-per-minute) and H4-Writer (6.83 words-per-minute). I finally demonstrate that with ThumbText, users can benefit from a unique text-entry technique that transfers well across different wearable displays, such as smartwatches and head-worn displays.

PUBLICATIONS

Some ideas and figures in this thesis have appeared previously in the following publications by the author:

- Junhyeok Kim, William Delamare, Pourang Irani. ThumbText: Text-Entry for Wearable Devices using a Miniature Ring. In *Proceedings of the 44rd Graphics Interface Conference (GI' 18)*, 2018.
- Junhyeok Kim, William Delamare, Yumiko Sakamoto, Tony Havelka, and Pourang Irani. Challenges Identified During Early Prototyping of a Ubiquitous Text-Entry System. In *Workshops of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17 EA)*, 2017.
- Junhyeok Kim, William Delamare, Yumiko Sakamoto, Tony Havelka, and Pourang Irani. Toward a Pool of Text-Entry Input Techniques. In *Workshops of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17 EA)*, 2017.

ACKNOWLEDGMENTS

I would like to take this opportunity to thank all the people who have helped me in the completion of this thesis.

I owe my deepest gratitude to my supervisor Dr. Pourang Irani for all the support and guidance not only for my research, but also as life advisor. This thesis would not have been possible without his expertise, assistance, patience, and encouragement throughout the process. His technical and editorial advice was essential to the completion of my research. I am honoured to have, as my committees, Dr. Carson Leung, and Dr. Zahra Moussavi who gave me insightful feedback on my thesis and advice on my future.

I am deeply grateful to Dr. William Delamare for his great advice. I have encountered many obstacles and troubles during my masters. I thank him for supporting this thesis. He always provided me with meaningful advice and thoughtful feedback.

I am thankful to Dr. Khalad Hasan for his genuine support, advice, and for finding time for me in his busy schedule. He has been always there for me since when I was an undergraduate student working on a summer project in the lab. He taught me innumerable lessons and insights on the workings of academic research in general. It is an honour for me to have Mr. Teng Han, who have always been there for me when I needed constructive criticism and feedback on my work. His support, guidance, and friendship were vital to my long journey of masters study.

I would like to show my gratitude to Dr. Yumiko Sakamoto for providing indispensable advice, information, and support on different aspects of my research. I would have struggled a lot more if it was not for her. I would like to thank Dr. Ahmad Bygowi for taking his time and effort to help me with the hardware aspect of my thesis. He has put his time and effort perfecting the device. I can not thank him enough for his kind help.

I want to acknowledge all the lab members, Zuoyi Zhang, Ali Neshati, Kenny Hong, Yurii Vasylkiv, Fouad Alallah who put their faith in me and urged me to achieve more. Much of my experimental work would have not been completed without their help.

My sincere gratitude to my family for continuous support and encouragement, and having patience and faith in me from miles away.

My special thanks to Dr. Barrett Ens, Dr. Gilles Ferrand, Dr. Juan David Hincapié-Ramos, Gaganpreet Singh, the staff members from the department of Computer Science and everyone else who inspired and helped me to go through this.

CONTENTS

LIST OF FIGURES

1

INTRODUCTION

Ubiquitous computing has been increasingly spotlighted in the last few decades due to its potential capability and advance in supporting technology [2], offering context awareness from embedded sensors. The concept of ubiquitous computing is that its user has access to information anytime and anywhere [16]. Recent display technologies allow various types of output and visualization techniques. However, these need to be used with suitable interaction techniques since it sometimes does not work well enough with traditional methods [2]. It is therefore necessary to offer display independent techniques. Since advanced display technologies offer novel opportunities to work with information almost everywhere, it is still an open question as to how and which interaction, specifically, input technique, should be used regarding user's preference and comfort. Much of this technology relies on direct touch input for each device including smartphones and smartwatches.

The steady miniaturization and improved functionality of wearable devices have made these technologies prominent among consumers' digital ecosystem. However, supporting common tasks, such as text-entry, is still an open challenge [41]. While wearables devices are primarily suited at displaying notifications, there are many instances that warrant entering text, such as to quickly reply to an SMS, for brief web queries, and for general input. Recent solutions have addressed some of the challenges for text-entry on smartwatches [13, 22, 24, 41] and head-worn-displays (HWDs) [21], the two more popular wearable platforms available today. While such approaches are optimized for each device, significant design iterations are needed to make them usable across devices. For example, novel design considerations were necessary for facilitating the transfer of a text-entry technique designed for smartwatches platform (SwipeBoard [13]) onto a HWD [21]. Offering a text-entry mechanism that works across multiple wearable devices provides the opportunity to design-once and reuse often. Such an approach mitigates concerns peculiar to any one device, such as finger occlusion on smartwatches [54], or fatigue [23] and social awkwardness on HWDs [21].

One promising solution is the use of wearable text-entry peripherals [18, 20, 28, 31, 34, 40, 48]. 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 [29, 49]. In other cases, the peripherals require a steep learning curve, making them unusable for short activity bursts [40, 53].

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 with one hand only
- self-contained: for optimum mobility, users should not be required to depend on additional materials or surfaces for text-entry
- unified input: users should be able to apply the same text-entry approach across wearable devices

As a result, my thesis designs, implements and evaluates a novel 'wearable-device' agnostic technique 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, this offers a one-handed indirect input, with subtle finger movements [12], and independent of the associated wearable display devices. Such a device can be used to operate smartwatches, HWDs, and possibly facilitate input during eyes and hands-busy operations.

Through a multipart design process, I first identify a suitable location for affixing a miniaturized touchpad onto the thumb's opposing fingers. I then assess the thumb's dexterity on this surface by carefully discretizing the available input space. The ideal discretization patterns is then used to map alphabet characters onto touchpad positions. While operating this new technique paired with a smartwatch, the technique outperforms existing state-of-the-art text-entry techniques. Lastly, minimizing performance loss between wearable devices is evaluated in a study involving novice users, and reveals that there is only minimal loss on performance between devices.

My contributions include:

- A text entry technique that can be used for multiple devices, such as smartwatches and head-worn displays
- A series of experiment validating the text entry technique
- Empirical results showing the thumb's performance on a small touchpad

My thesis is structured as follow:

1. Motivations and challenges
2. Literature review - discusses existing text-entry hardware and software
3. Apparatus for the experiments in this thesis
4. Investigation about the relationship between the human thumb and a small touchpad
5. Validation and Evaluation of the technique
6. Conclusion and Future works

2

MOTIVATION

My goal is to design a text-entry approach that can be used across wearable devices. The main benefit of such work resides in the fact that all wearable displays can use the same text-entry technique and input device, minimizing the transfer cost when switching from one device to another. Additionally, one can imagine near-future wearable displays such a contact lenses [33] having limited input capabilities, thus benefiting from methods such as the one I propose. I describe the requirements leading up to my design rationale and the challenges related to the goal of this work.

2.1 REQUIREMENTS

I identify the following common requirements based on the context in which wearable devices are used. Since I focus on text-entry for any wearable device, my input mechanism needs to accommodate indirect input, i.e. where the input space is separated from the display. Furthermore, the use of wearable devices for mobility implies scenarios in which the users' hands are busy, like holding a bus handle. Hence the device needs to operate in one-handed modes. As both the device and the input technique need only one hand, the device needs to be worn, without being actively held. Finally, mobility also suggests use in varying contexts, thus requiring resistance to environment noise (e.g., sound or light). The above criteria eliminate a number of devices, such as chording input devices or one-handed keyboards [34] as these need to be actively held, and sound-based input [47] as this is not resistant to environment noise. I avoid mid-air finger gestures as such interaction can cause fatigue and is a source of jitters [15]. Based on the above requirements, I settle on using a ring form factor as it easily satisfies all these requirements. Other wearable form factors are also possible, however, a ring is socially acceptable, always accessible and convenient [4, 30] as a wearable device. Using the ring as a touchpad is a good way to provide a form-factor adapted for finger input [56] and to minimize the input footprint, i.e. the amount of area consumed by the input [9]. To do so, I want to maximize the bandwidth of two fingers interaction [6]: as fingers working together offer a better information processing rate (4.5 bits/s for pinch gestures) than other body parts such as the wrist (4.1 bits/s) or a finger alone (3.0 bits/s). Thus, my goal is to design a text-entry for any wearable display, using a touch-sensitive ring form-factor involving two fingers: the finger wearing the ring and its opposing thumb.

2.2 CHALLENGES

Providing a ubiquitous text-entry technique on a ring touchpad presents numerous challenges. These challenges are related to both the input and display space. A touchpad affixed to a ring needs to be small for comfort and social acceptability. Thus, the input space will be limited. The standard deviation of an eye-free tap gesture performed with the thumb on a mobile phone is X=2.59mm and Y=4.05mm [50]. This means that on a 1cm x 1cm touchpad, I can theoretically place $1cm^2$ / (0.259cm × 0.405cm): approxmiately 9 items that could comfortably be selected eyes-free. Human resolution can go up to 0.17mm when the index finger is directly in contact with a surface [8]. However, it is not clear if this result applies to thumb input with a different form-factor such as the ring touchpad where input and output spaces are separated. My first experiment shows that users can select up to 8 items with the thumb on cells as small as $0.26cm^2$ on a 1.8cm×1.3cm ring touchpad. My device should allow users to perform text-entry tasks on any wearable display. Devices such as smartwaches and smartglasses offer relatively small displays. Thus, the interaction technique should consider a simple and minimalistic virtual keyboard layout. My second experiment explores the effect of the graphical design of the keyboard layout. My third and fourth experiments evaluate my technique in different difficulties (e.g. entering less frequent English characters). My last experiment explores the effects of different output devices on my final technique, namely ThumbText. Results show that ThumbText accommodates text-entry well on two common wearable displays, the smartwatch and smartglasses.

3

LITERATURE REVIEW

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

3.1 PERIPHERAL DEVICES AND APPLICABILITY

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

3.1.1 *Handheld Devices*

Twiddler [34] is a one-handed keyboard device that allows user to input characters. Text-entry rates can reach 60 word per minute on average. However, Twiddler requires user to hold a physical device on one hand, preventing users to perform other task while holding it. GesText [28] uses accelerometer based gestures for text entry by defining a number of directional gestures to enter characters using only accelerometer data. However, this requires users to hold the device in one hand. Handheld devices do not satisfy my miniaturization design requirement.

3.1.2 *Wristband Devices*

One-Key Keyboard [31] uses a touch-sensitive surface attached to the wrist for text-entry. The prototype provides haptic feedback to produce a touch experience similar to mechanical keyboards. PalmType [48] uses an array of IR sensors embedded on a wristband to detect finger touch on the palm. PalmType allows users to input characters by projecting a virtual keyboard onto the palm. Touch-Sensitive Wristband [18] enables touch interaction directly on the wristband. This solution avoids using on-screen space to display the keyboard. However, these wristband solutions force users to interact with both hands: one for wearing the wristband, one for typing. Therefore, this does not meet my one-handed requirement.

3.1.3 *Ring Devices*

TypingRing [40] is a text-entry technique which allows users to type letters with three fingers on any solid surface. Similarly, TAP [26] allows users to type text with chorded input on a near-by surface. Both TypingRing and TAP 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.

3.2 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 [52] allows users to input letters by performing sequences of hits in the corners of a square, offering improved entry-rates compared to Graffiti [10]. However, input vocabularies [10, 52] require users to pre-learn gestures, an often heavy burden for pick-up-and-use contexts. 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 [19], InvisiBoard [38], or COMPASS [55]. 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 [19]. Interaction-based techniques propose to select a letter via an explicit disambiguation step using a second modality (e.g., Tilt-Text [51], TiltType [42] or GesText [28]), such as a wrist rotation. In this work, I focus on techniques using only one modality: one finger touch.

3.3 MULTI-STEP TEXT-ENTRY TECHNIQUES

I distinguish between techniques focusing on soft keyboard manipulation, and techniques focusing on cluster of characters.

First, text-entry techniques can propose to manipulate the soft keyboard visualization. With ZoomBoard [41], the area surrounding a first tap position 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 enough when the input and output spaces are decoupled, as with indirect input. ZShift [32] 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 [11] 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 [24] displays only one half of a QWERTY keyboard at a time. Users can swipe horizontally to switch between each half-keyboard. DriftBoard [45] allows users to pan a soft keyboard to select characters via an on-screen pointer at a fixed location. The indirection touch/pointer overcomes the fat-finger problem.

Second, text-entry techniques can propose multiple characters per area followed by a disambiguation step. For instance, with Quikwriting [43], 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 [1] builds on Quikwriting and allows users to choose which subgroup they want to select a character from by moving their finger in a clockwise or counter-clockwise direction. Several other techniques adopt the same strategy with different discretization layouts. With MessagEase [39], users perform a first tap to select a cluster of characters, followed by a slide and lift action to further refine the selection. Other techniques build on MessagEase by using different layouts [7], by reducing the number of characters in each area [44], or by integrating a predictive mechanism [17]. The H4 family, e.g., H4-Writer [36] and H4-TEG [5] 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, I hypothesize that this technique can also be adapted on a small ring touchpad. With SwipeBoard [13], users perform two swipe gestures 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 [21] extends the SwipeBoard technique to consider the specific form factor of Google Glass' lateral touchpad. Authors chose to base their technique on 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 specifically for an indirect ring form-factor for text-entry on wearable devices. In the remainder of this paper, I present my design process for ThumbText.

4

APPARATUS

I describe my ring prototype, and the hardware and software apparatus used in my studies.

4.1 RING PROTOTYPE

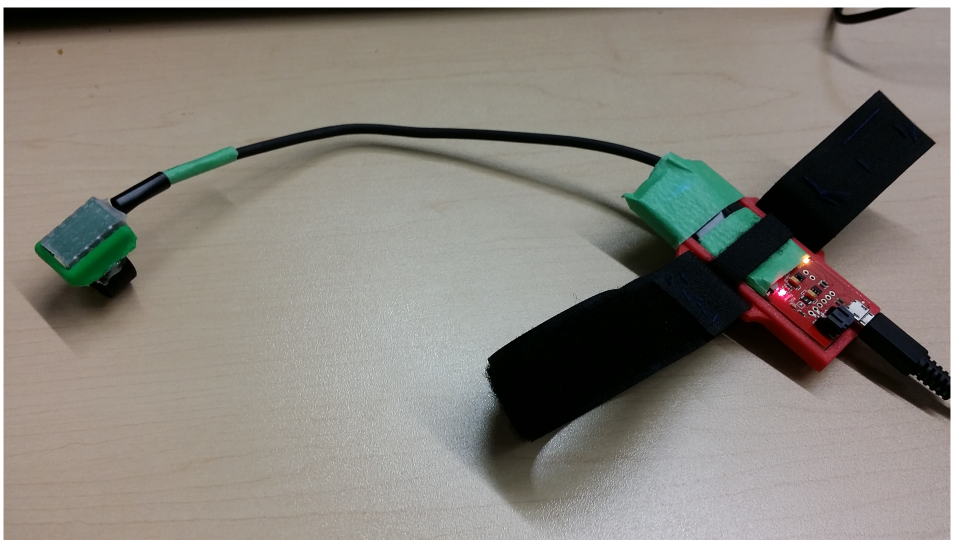


Figure 4.1: The prototype used

The ring prototype (4.1) consists of a MTCH6102 Low-Powered Projected 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 prototype device communicates with an Arduino Fio V3 via the I2C protocol.

4.2 SOFTWARE

For every studies in this thesis, the main software is implemented in C# with the Unity3D 5 game engine. The software is able to detect basic touch surface gestures, including tap and swipes in different directions. I did not use built in functions that is built in the micro-controller to avoid latency and have better control over the specific gesture set for the experiments.

4.3 SETTINGS

Figure 4.2: Smartwatch: iMacwear M7

All of the studies were conducted in a controlled setting in lab. The custom software ran on a 3.4 GHz Intel Core i7-3770 with 8GM RAM

Figure 4.3: Smartglasses: Epson Moverio BT-200

computer. The ring prototype (i.e. Arduino board) communicated with the main software via USB. Participants were sitting approximately 50cm away in front of a Dell U2312HM with 1920x1080 resolution desktop monitor. Wearable display devices were an iMacwear M7 smartwatch(4.2) and the Epson Moverio BT-200 smartglasses (4.3). Data communication between the main software and the ring (resp. Wearable displays) was done via USB (resp. Wi-Fi).

5

INVESTIGATION OF HUMAN LIMITATION

5.1 EXPERIMENT 1: PRECISION

My objective is to get users to input more than 28 characters. Thus, in the first experiment, I explore the performance of thumb input on a ring touchpad (i.e. Precision of the thumb using touchpad).

5.1.1 *Factors*

I consider three experimental factors: the shape used to discretize the touchpad for selection purpose, the level of discretization and the finger to which is attached the ring touchpad.

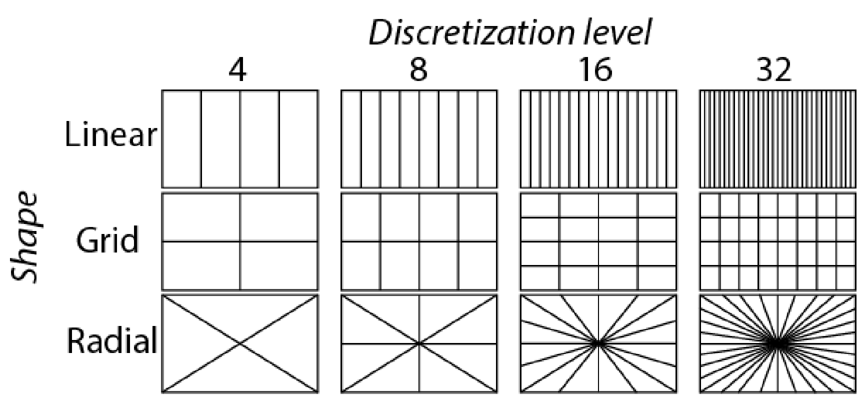


Figure 5.1: *Discretizations* and *Shape*

DISCRETIZATION SHAPE I consider three discretization shapes: linear, grid and radial.

- Linear: A linear discretization is the simplest form (5.1). I choose the horizontal discretization, adapted to the wider form factor of our prototype. The linear shape is meant to prevent errors from vertical motions. However, the linear discretization can limit the number of unique areas.
- Grid: A grid-shaped discretization divides the touchpad into rectangular areas (5.1). A grid offers larger areas than the linear discretization for horizontal precision, but requires users to be precise in the vertical direction.
- Radial: Finally, a radial-shaped discretization divides the touchpad into triangular areas radiating from the touchpad center (5.1). Compared to the grid-shaped discretization, a radial layout offers larger selectable areas near the edges of the touchpad which gives users haptic feedback when performing eyes-free selections.

DISCRETIZATION LEVEL : I also explored the number of areas that could be defined on the touchpad. I consider discretization levels of 4, 8, 16 and 32 areas. Levels of the linear discretization are equal to the number of columns. Levels of the grid discretization lead to 2×2, 2×4, 4×4 and 4×8 grid sizes (5.1). I choose more columns than rows to fit our prototype wider form-factor. Levels of the radial discretization are equal to the number of triangles, with all areas occupying the same surface area.

FINGER : I consider two fingers on which to attach the ring touchpad: the index and the middle finger.

- Index Finger: The index finger is the first finger opposite to the thumb: the thumb can hence reach the touchpad quickly and easily.
- Middle Finger: The middle finger offers the advantage to keep the index finger free while still allowing a comfortable access to the touchpad. Keeping the index free is a desirable property to offer additional input in the future like pointing at a distant display or tapping on the ring to switch between lower-case and upper-case modes.

5.1.2 *Participants*

A total of 12 participants (5 females), between 19 and 35 years old (M=23.16, SD=4.08), were recruited from the local community. 11 of them were right-handed and one was ambidextrous. 11 participants had 3+ years experience with touch sensitive devices, including smart phones and tablets. Finally, 5 participants had experience with wearable devices such as smartwatches and smartglasses for over a year.

5.1.3 *Procedure*

The experiment lasted 30 minutes per participants after which they filled a questionnaire for qualitative data. The task consisted in selecting a coloured target displayed on a screen in front of the user. I asked participants to not look at the ring touchpad or their fingers to assess the precision during eyes-free selection. They were instructed (i) to be as accurate and as fast as possible, and (ii) to wear the ring on their finger's middle phalanx. Participants could practice for a minute before each condition. The target to select was colored in blue. Participants validated the target selection by lifting the thumb from the ring touchpad. Participants were instructed to select a target by tapping on the touchpad as close as possible to the target. If they needed to correct the landing position, (1) the current 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.

5.1.4 *Experimental Design*

I 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 phases: one for each finger. Each phase contained three blocks, i.e. one block per technique. Each block consisted in 4 series of trials, one per discretization level. For each condition, I 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.

5.1.5 *Experiment 1: result*

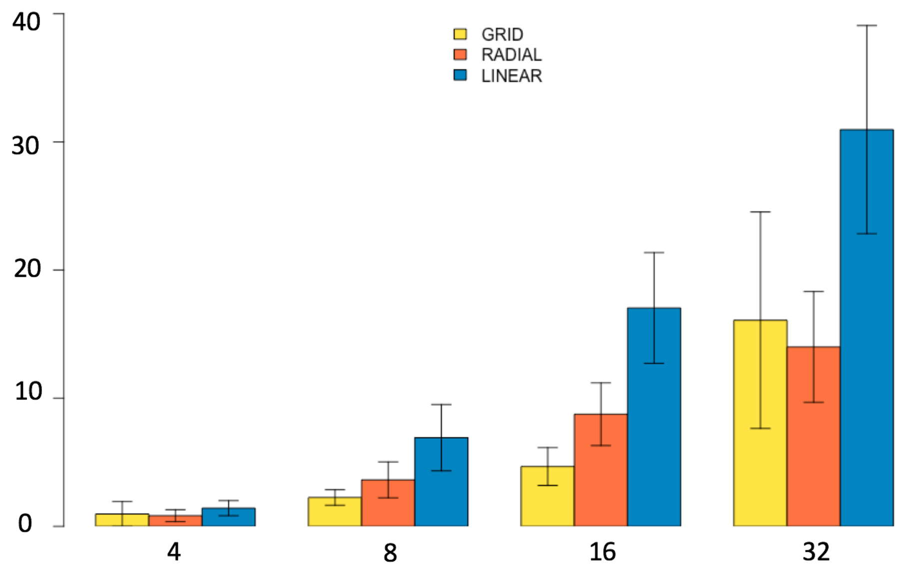
The main dependent measures for the task were the selection time and the error rate defined as:

$$ER = \frac{\text{Number of Errors}}{\text{Number of Success} + \text{Number of Errors}}$$

I collected a total 8340 trials, including 5760 successful selections and 2580 failed selections.

5.1.5.1 *Error Rate*

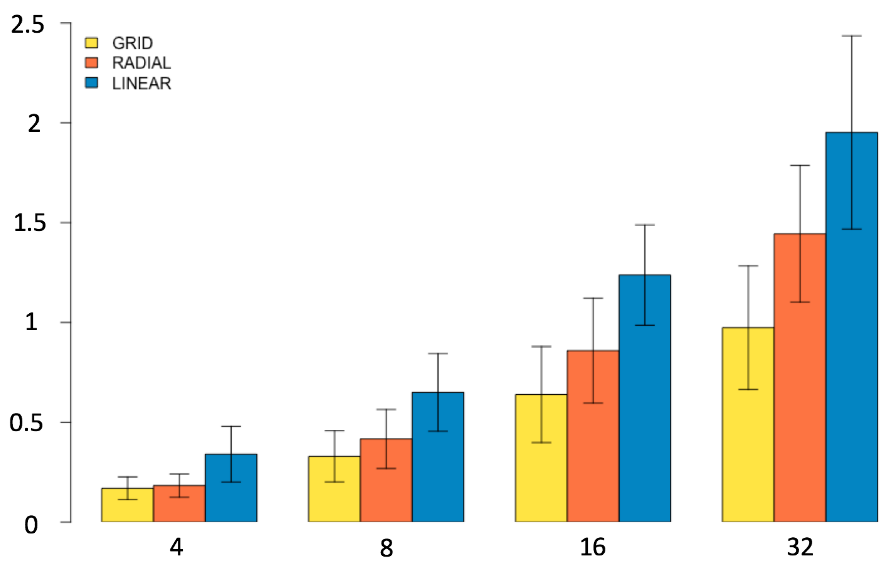
I found a significant effect of *Shape* on the error rate [F2,26=49.81, p<0.001, ηp²=0.82]: Grid offers a better accuracy than radial [p<0.01], which gives a better accuracy than linear [p<0.001]. I also found a significant effect of *Level* [F3,26=70.23, p<0.0001, ηp²=0.86] with each level being significantly more error prone than the previous smaller one [p<0.001]. I did not find any significant effect of *Finger* on the error rate [F1,26=4.32, p=0.06]. The grid and radial shapes are significantly different only for the discretization level 16 [F2,33=22.04, p<0.001, ηp²=0.67] (5.2). In contrast, the linear shape is significantly different from the two other shapes as soon as level 8 [F2,33=15.85, p<0.001, ηp²=0.59].



(a) Error Rate (%)

Figure 5.2: Experiment 1 results

5.1.5.2 *Selection Time*



(a) Selection Time (sec.)

Figure 5.3: Experiment 1 results

I consider only successful trials for the selection time analysis. I applied a log transform to my data to satisfy the normality and homogeneity of variances assumptions. Figures show non-transformed data. I performed two-ways ANOVAs and accounted for repeated measures by treating participants as a random variable. I used multiple pairwise t-test comparisons with a Bonferroni correction for post-hoc tests.

I found a significant main effect of *Shape* [$F_{2,26}$=48.49, $p<0.00001$, ηp^2=0.82] and *Levels* [$F_{3,26}$=96.97, $p<0.00001$, ηp^2=0.90], but no effect of *Finger* [$F_{1,26}$=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 leads to faster selection times than linear. I also found significant differences between all discretization *Levels* [$p<0.001$]: the more discretized areas, the longer the selection time (5.3). I did not find any interaction effect.

5.1.6 *Discussion*

From the results, I infer the need for a two-step interaction technique to input 28+ characters on my 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 (5.4, A), I focus on two-step interaction techniques using the grid-shaped discretization. Regarding the finger to which the ring touchpad is at-

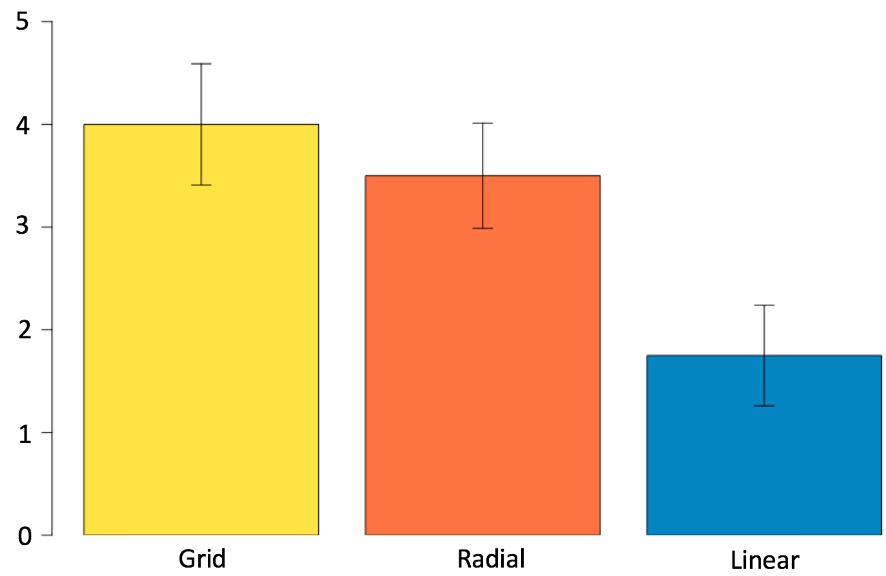


Figure 5.4: Preference on Shape

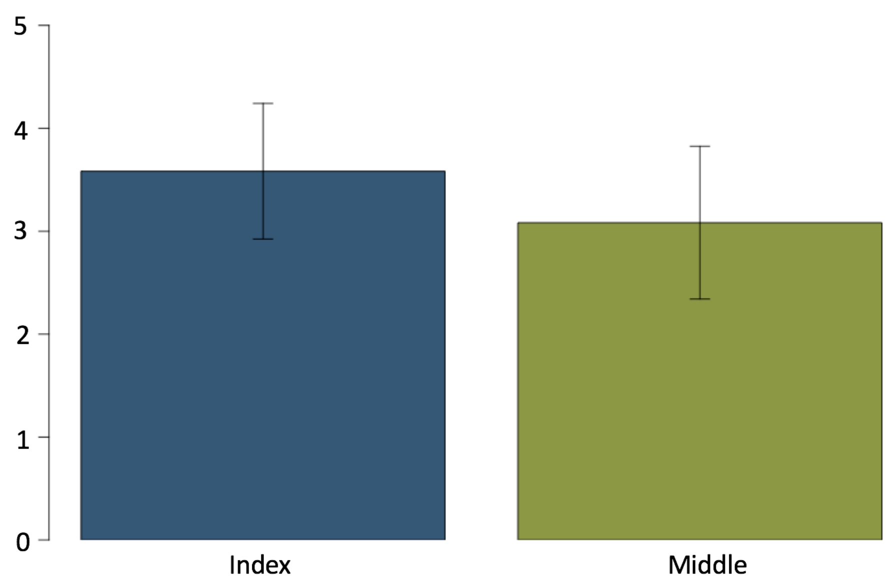
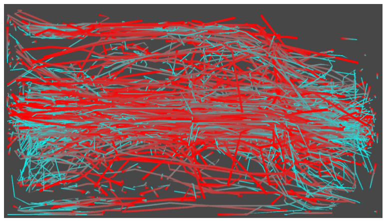


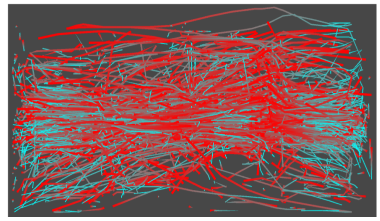
Figure 5.5: Preference on Finger

tached, I did not find any significant difference. However, it seems that participants preferred using the ring on the index finger (5.5, B). I hence use the index finger for the remaining studies.

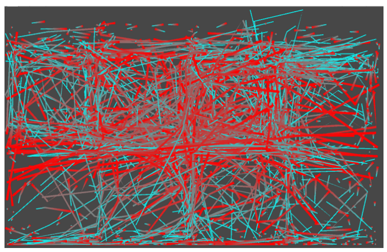
5.1.6.1 *Visualization of finger movement*



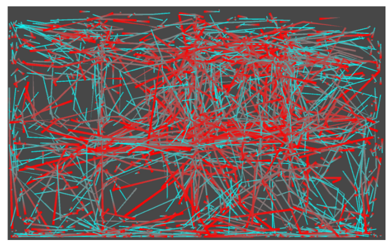
(a) Visualization of Linear with index finger



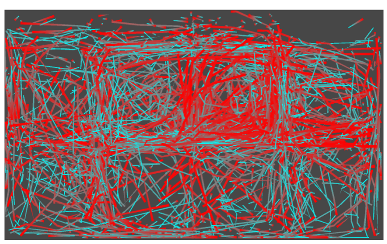
(b) Visualization of Linear with middle finger



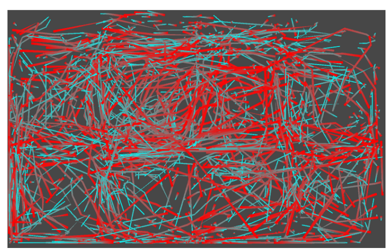
(c) Visualization of Grid with index finger



(d) Visualization of Grid with middle finger



(e) Visualization of Radial with index finger



(f) Visualization of Radial with middle finger

Figure 5.6: Visualizations of Finger trail

Here are the visualizations of finger movements tracked during the experiment. Each stroke represents a single target selection sequence, i.e finger landing on the touch surface, finger moving for correction, and finger lifting for validation. In these visualizations, landing position is coloured in red and lifting position is coloured in cyan, with a gradient in-between to display the finger movement. Each shape visualization shows that the finger trail follows the cell shape. For example, participants moved their finger linearly on the Linear shape and with rectangular pattern on the Grid shape. This indicates that the shape of the discretization might force the finger movement to follow certain patterns.

5.2 EXPERIMENT 2: TECHNIQUE DESIGN

My approach to increase the number of selectable areas on a ring touchpad involves a two-step concept: a top-level for selecting a cluster of letters, and a lower-level for selecting a single character.

5.2.1 *Factors*

LAYOUTS I consider a way to increase the number of selectable areas on a ring touchpad: involving a two-step approach: a top-level and nested within a lower-level.

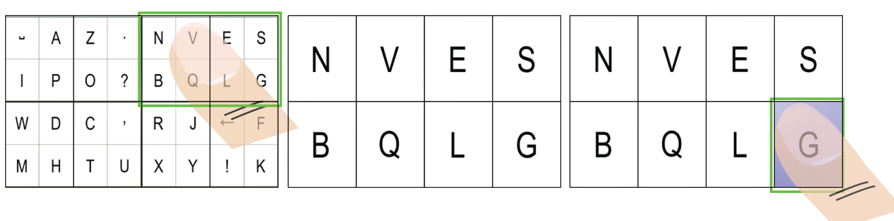


Figure 5.7: Tap interaction using $Grid4 \times Grid8$

- Grid4-Grid8: I first consider the simplest grid shaped discretization: 4 areas. The second step can use a discretization of 8 areas, offering $4 \times 8 = 32$ characters (5.7). Once a top-level area is selected, the associated characters are displayed in a grid-8 design.

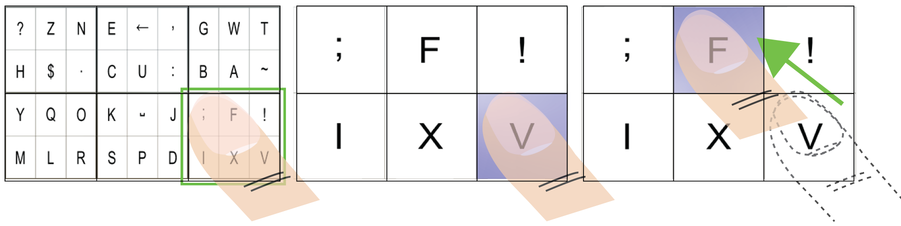


Figure 5.8: Touch and Lift interaction using *Grid*6 × *Grid*6

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: I 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 $6 \times 6 = 36$ characters (5.8).

INPUT TECHNIQUES For both Layouts, I 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 (5.7). I 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 move their thumb on the touchpad to navigate in

the lower-level, and (iii) validate their selection by lifting their thumb from the touchpad (5.8). 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. I designate these designs using the 'lift' keyword and the discretization used, i.e. Lift-4-8 and Lift-6-6.

5.2.2 *Participants*

8 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.

5.2.3 *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 colouration 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.

5.2.4 *Experiment 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*. I then considered two distinct repetition blocks with 5 successive sequences of trials each. The independent variables were the *Design* and the *Repetition* (1, 2). I collected 4 designs × 8 characters × 10 repetitions = 320 acquisitions per participant, for a total of 3200 acquisitions.

5.2.5 *Result*

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

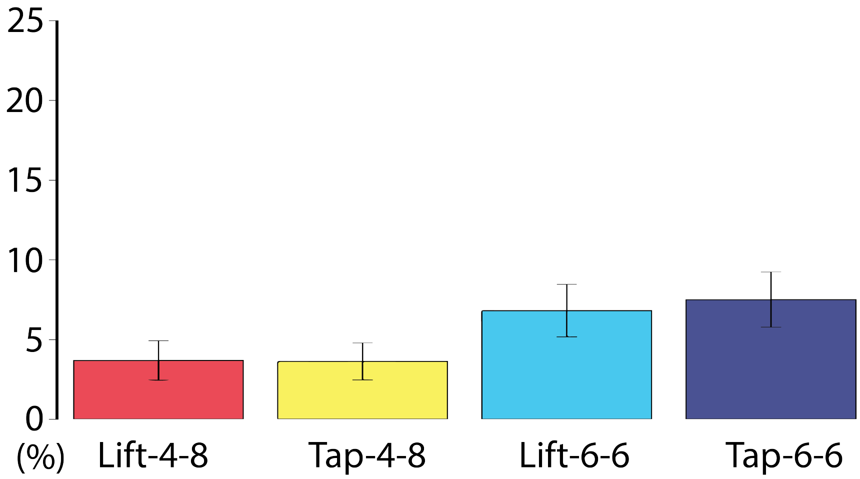


Figure 5.9: Experiment 2 Soft Error

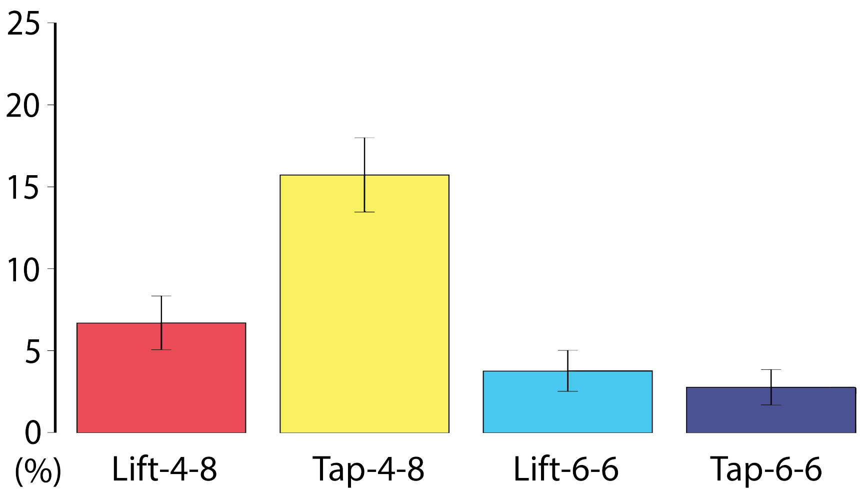


Figure 5.10: Experiment 2 Hard Error

5.2.5.1 *Error Rate*

I 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*[$p<0.05$]. I 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 [$F_{3,56}=2.64$, $p=0.08$](5.9). For the hard error type, I 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] (5.10).

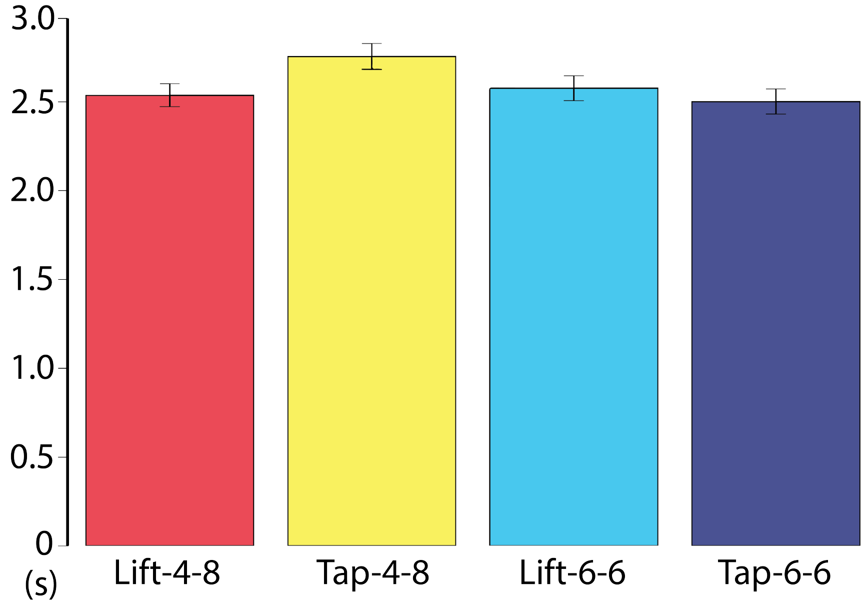


Figure 5.11: Experiment 2 Selection Time

5.2.5.2 *Selection Time*

I performed two-way ANOVAs on log-transformed successful data and accounted for repeated measures by treating participants as a random variable. I used multiple pairwise t-test comparisons with a Bonferroni correction for post-hoc tests. I did not find any significant effect of *Design* on the selection time [$F_{3,56}=2.23$, $p=0.12$] (5.11). There was a learning effect [$F_{1,56}=88.41$, $p<0.0001$, $\eta p^2=0.93$]: participants were faster at the end than at the beginning with all *designs*. There was no interaction effect.

5.2.6 *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 (5.8, 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, I chose Lift-6-6 as a basis for ThumbText.

6

EVALUATION OF THE TECHNIQUE

6.1 EXPERIMENT 3.A: TECHNIQUE EVALUATION IN DIFFICULT CONTEXT

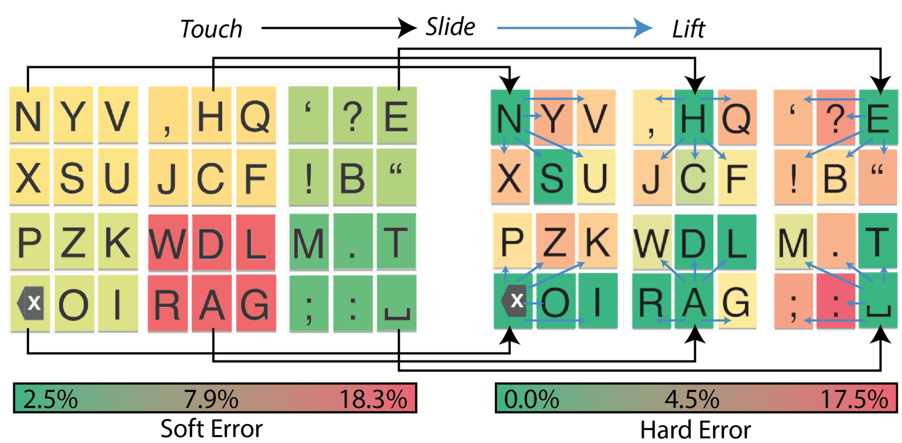


Figure 6.1: Heatmap based on error rate

I generated a heatmap (6.1) 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). I explore how Thumb-Text performs under difficult conditions against existing techniques, namely SwipeBoard [13], and H4-Writer [36]. My 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 my ring touchpad based on results from experiment 1. SwipeBoard uses swipe gestures to select characters. I hence want to evaluate how Tap (H4-Writer), and Touch-and-lift (ThumbText) perform against swipe gestures that is not considered in the previous experiments. The study uses my 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's original, and into a 4-cells radial layout for H4-Writer. I adapted the protocol from previous work to simulate expert performances [13] in a reasonably short time. I 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 for each of the technique. This also allows to determine the lowest performance of my technique: any other context will hence lead to similar or better text-entry rates. I 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 (6.1). 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.

6.1.1 *Experiment Design*

The task used a repeated-measure within-participant design with *Technique* (ThumbText, SwipeBoard, and 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 trials containing one word randomly picked in the set of 17 words generated via my 6 letters. I collected 2 sessions × 3 *Technique* × 4 *blocks* × 5 sets × 7 trials = 840 words per participants, for a total of 7560 acquisitions.

6.1.2 *Participants*

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

6.1.3 *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 word to enter 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 I wanted to evaluate the learning process.

6.1.4 *Result*

I report standard text-entry metrics when applicable [3, 46]. 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 percent. I focus my analysis on three quantitative metrics (words-per-minute, total error rate, and keystroke per character) and on qualitative feedback. Each metric is calculated with the following formulas [3]:

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

$$TotalER = \frac{UnnoticedError + CorrectedError}{|Corrected| + UnnoticedError + CorrectedError} \times 100\%$$

$$KSPC = \frac{|InputStream|}{|TranscribedText|}$$

- *Transcribed Text* is the final input text entered by the participants.

- *Unnoticed Error* is the number of unnoticed errors in *Transcribed Text*.
- *Corrected Error* is the number of noticed error not appear in *Transcribed Text*.
- *Corrected* is the characters entered by the participants correctly.
- *Input Stream* is the text containing all keystrokes including editing keys.

I 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.

6.1.4.1 *Words-per-minute - WPM*

I found a significant main effect of *Technique* [$F_{2,18}=14.64$, $p<0.0001$, $\eta p^2=0.65$] and *Block* [$F_{7,18}=35.06$, $p<0.0001$, $\eta p^2=0.81$] 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. I didn't find interaction effect between *Technique* and *Block*.

6.1.4.2 *Total Error Rate - Total ER*

I did not find any significant effect from *Technique* on the total error rate [$F_{2,18}=2.27$, $p=0.13$]. I found a significant main effect of *Block* on the total error rate [$F_{7,18}=3.47$, $p<0.005$, $\eta p^2=0.30$], as participants managed to decrease the number of errors from 11.96±7.08% in the

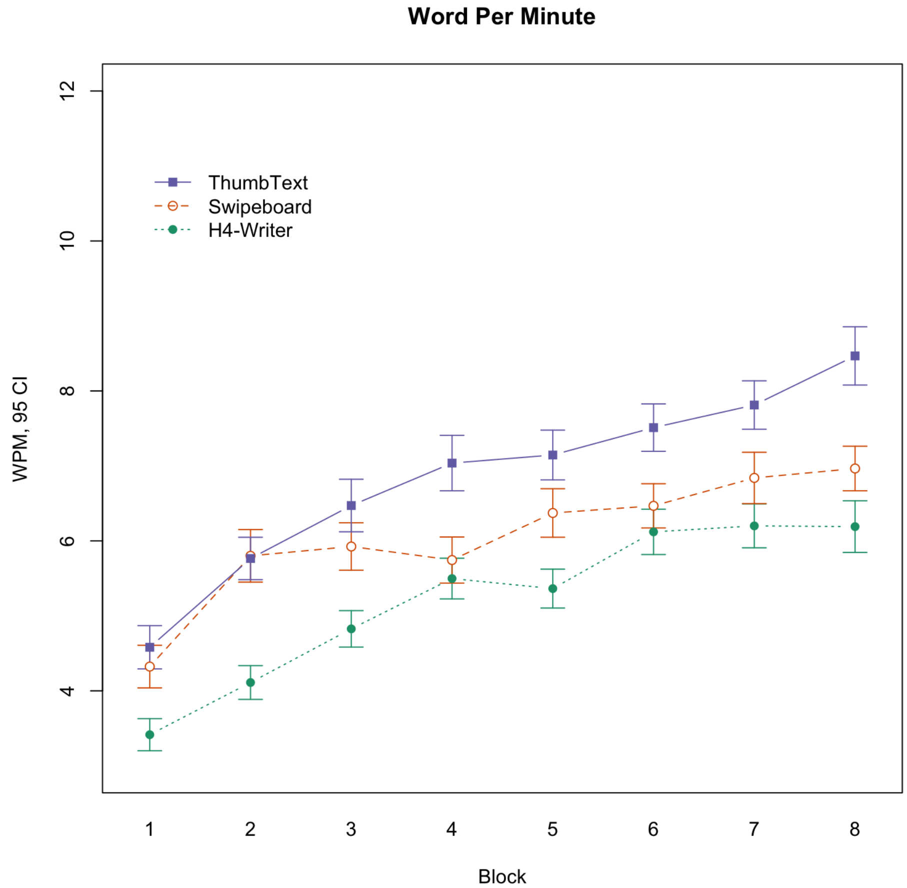


Figure 6.2: Words-per-minute of each technique

1st block to 7.94±3.96% in the last block. I also found an interaction effect between *Technique* and *Block* [$F_{14,18}$=2.25, $p<0.05$].

6.1.4.3 *Keystrokes Per Character - KSPC*

I found a significant main effect of *Technique* [$F_{2,18}$=25.77, $p<0.0001$, ηp^2=0.76] and *Block* [$F_{7,18}$=6.30, $p<0.0001$, ηp^2=0.44] on the KSPC. I also found an interaction effect between *Technique* and *Block* [$F_{14,18}$=2.94, $p<0.005$, ηp^2=0.26]. With SwipeBoard, KSPC reaches a plateau after the 3rd block around 2.75±0.61. Nearly constant KSPC values indi-

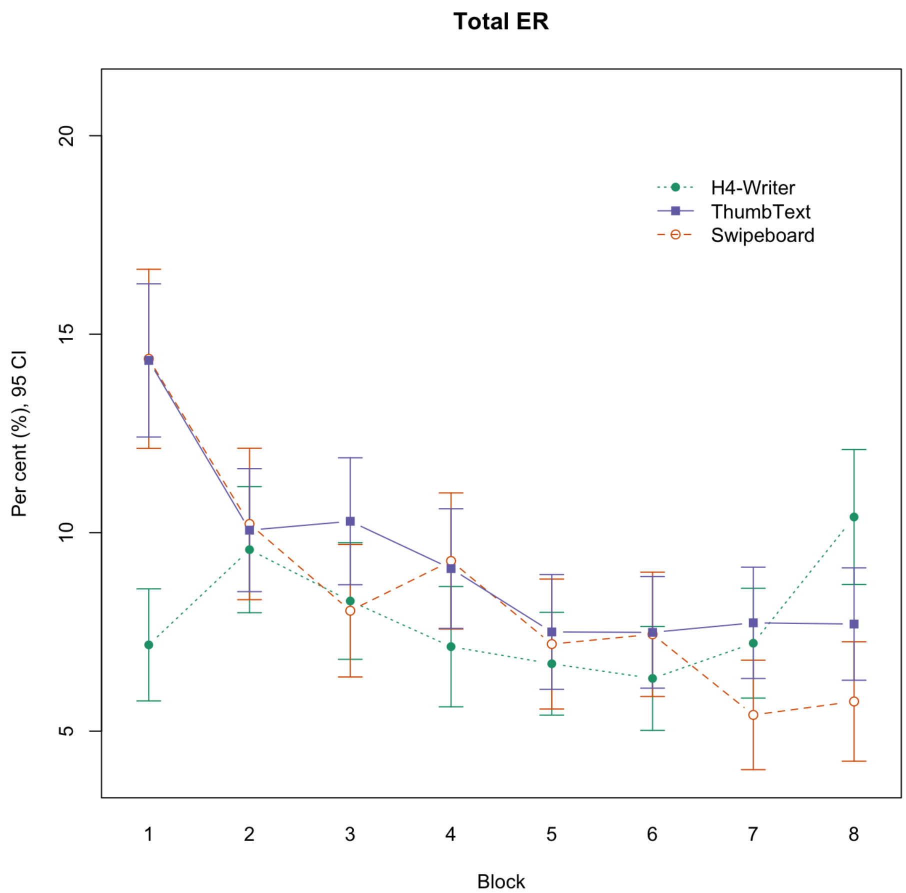


Figure 6.3: Error rates of each technique

cates 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.

6.2 EXPERIMENT 3.B: TECHNIQUE EVALUATION IN NORMAL CONTEXT

I next evaluate ThumbText without focusing on characters explicitly difficult to input. For this, I choose characters based on H4-Writer.

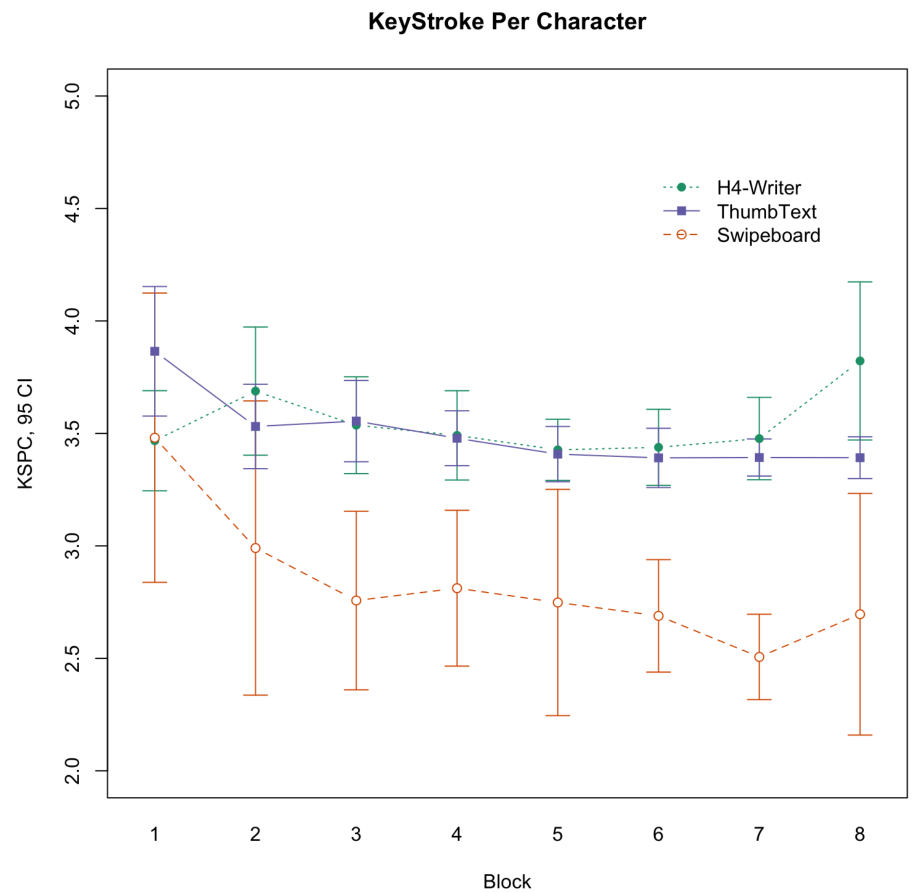


Figure 6.4: Keystroke per character of each technique

H4-Writer optimizes the complete set of characters based on their frequency. Evaluating techniques with characters spread out along this frequency range allows to evaluate techniques in a normal context (i.e. not designed to be difficult or easy). In this comparison, I 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 to evaluate the techniques 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. I expect H4-Writer's performance to incur a drop of performance compared to the previous study, and SwipeBoard's performance to remain the same.

6.2.1 *Participants*

I recruited new 9 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 (including smart wristband). The experiment followed the exact same protocol and procedure as Experiment 3a.

6.2.2 *Result*

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

6.2.2.1 *Words-per-minute - WPM*

I found a significant main effect of *Technique* [$F_{2,18}=35.96$, $p<0.0001$, $\eta p^2=0.82$] and *Block* [$F_{7,18}=130.74$, $p<0.0001$, $\eta p^2=0.94$] 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)

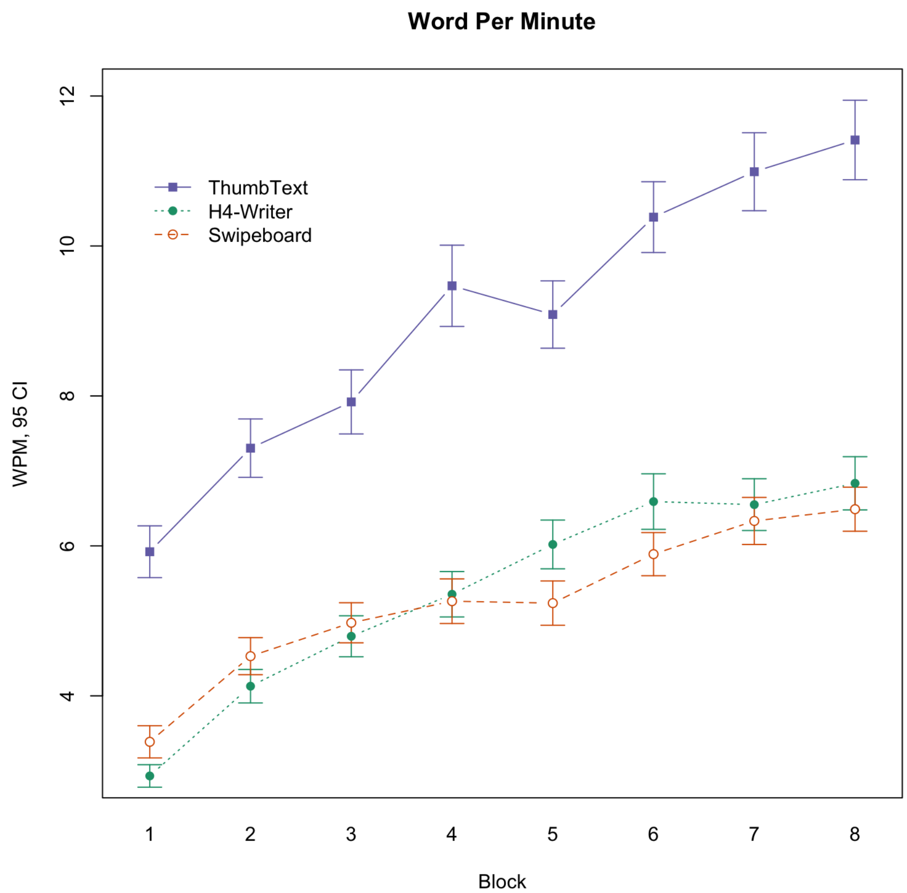


Figure 6.5: Words-per-minute of each technique

[$p<0.0001$]. I also found an interaction effect [$F_{14,18}=1.80$, $p<0.05$, $\eta p^2=0.18$]: 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 [$p<0.01$].

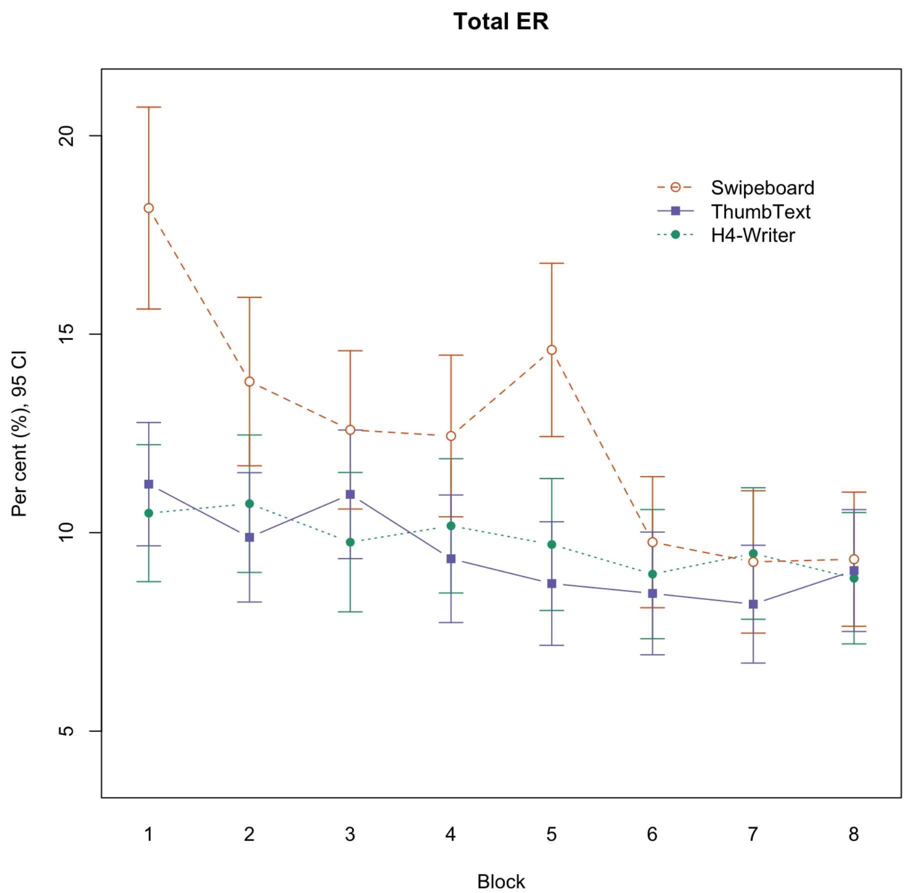


Figure 6.6: Error rates of each technique

6.2.2.2 *Total Error Rate - Total ER*

As in the previous experiment, I did not find any significant effect from *Technique* on the total error rate [$F_{2,18}$=3.51, p=0.055]. I found a significant main effect of *Block* on the total error rate [$F_{7,18}$=2.91, $p<0.05$, ηp^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. I 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.

6.2.2.3 *Keystrokes Per Character - KSPC*

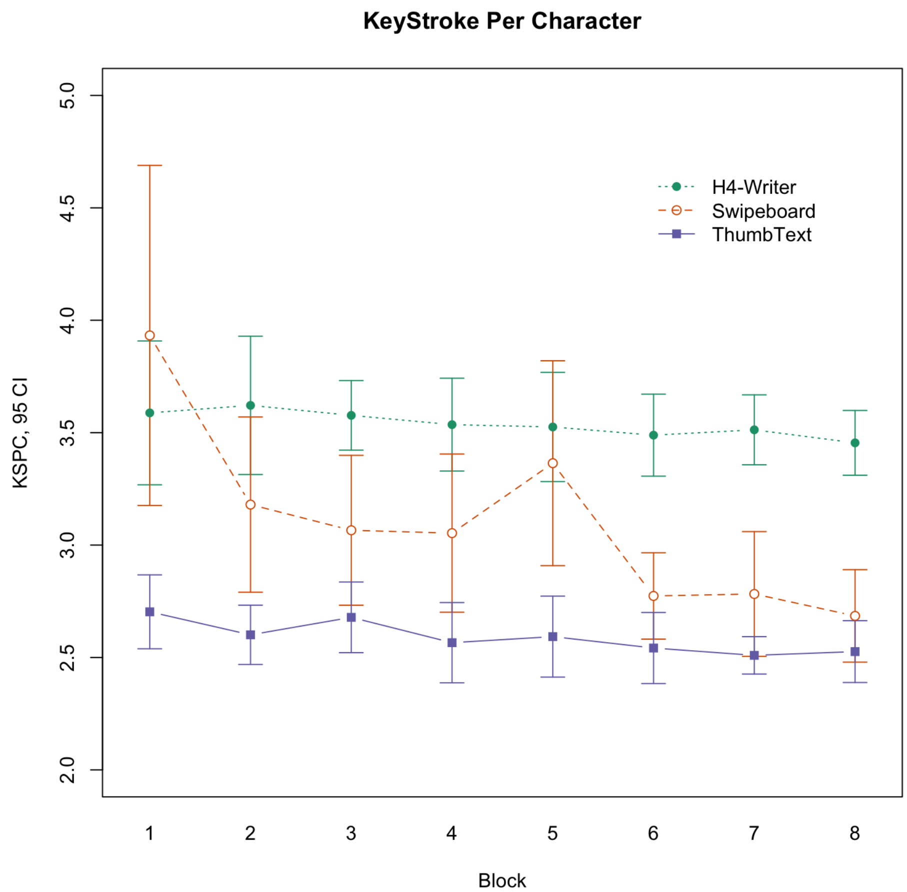


Figure 6.7: Keystrokes Per Character of each technique

When using ThumbText and H4-Writer, participants increased their text-entry rate without significantly decreasing their error rate. Constant KSPC values support this observation ($p>0.05$). 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.

6.3 DISCUSSION

I discuss the performance measures according to both studies: difficult and normal contexts.

6.3.1 *WPM*

Both cases, 8.46±2.45 wpm (experiment 3.a - difficult context) and 11.41±2.30 wpm (experiment 3.b - normal context) show that my 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.

6.3.2 *Total ER*

From the results, I 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 my initial hypothesis.

6.3.3 *KSPC*

ThumbText dropped its KSPC values from experiment 3a (3.86 in the first block, 3.39 in the last block) to experiment 3b (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.

6.4 CONCLUSION

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

Second, I 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 [45]. Finally, qualitative results show that participants reported that SwipeBoard's QWERTY layout was an asset compared to the other techniques (P3, P4, P7, and 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, and P7) and the number of taps to enter a character (P3, P4, and P7). Participants also noticed the unusual layout used by ThumbText (P2, P6, P7, and P8). However, participants considered

ThumbText as fast, convenient (P1, P3, P7, P9, P1′, P2′), and easy to use (P1, P5, P8). Two participants reported some fatigue while using ThumbText (P2, P6). Overall, participants preferred ThumbText and H4-Writer to SwipeBoard.

6.5 EXPERIMENT 4: DEVICE EFFECT

In my final experiment, I study (1) how ThumbText performs with different wearable displays, and (2) to compare ThumbText to a baseline. I choose SwipeBoard[13] as (1) it shows good performances for non-predictive smartwatch input, and (2) it uses swiping gestures that meet our requirements, i.e. one-handed modes as well as being usable on our small ring touchpad.

6.5.1 *Participants*

12 participants (7 females), aged between 21 and 31 (M=23.7, SD=2.8) volunteered for the experiment. All of them had more than 3 years of experience with touch sensitive devices and one participant had more than 3 years of experience with wearable devices (smartwatch).

6.5.2 *Procedure*

The experiment lasted on average 75 minutes per participant after which they filled a questionnaire. The task in the experiment consisted in inputting characters to transcribe full sentences. The sentence to transcript was displayed on top of the display. The input sentence was displayed as participants entered characters in a textbox positioned below the presented sentence. Participants could rest between sentences.

6.5.3 *Experimental Design*

The independent variables were the *Device* (smartwatch, smartglasses) and the *Technique* (ThumbText, SwipeBoard). The ordering of *Device* and *Technique* was counterbalanced between participants using a Latin-square design. The experiment was divided into two sections (one for each *Technique*). Each section consisted in two blocks (one per *Device*). In each condition, participants had to enter 2 practice sentences followed by 12 sentences. Participants had to input two 'space' characters in order to validate the trial. Sentences were randomly taken from the sentences set from Mackenzie et al. [35]. I made sure that these random sentences had a maximum length of 5 words in order to keep the experiment reasonably short (100 minutes). This experimental design resulted in 2 Device × 2 Technique × 12 sentences = 48 sentences per participants, for a total of 576 sentences.

6.5.4 *Result*

My dependent variables are the trial time and the error rate. I refine analysis with the standard text-entry metrics found in the literature [3, 46]. I hence present these error rate measures along with

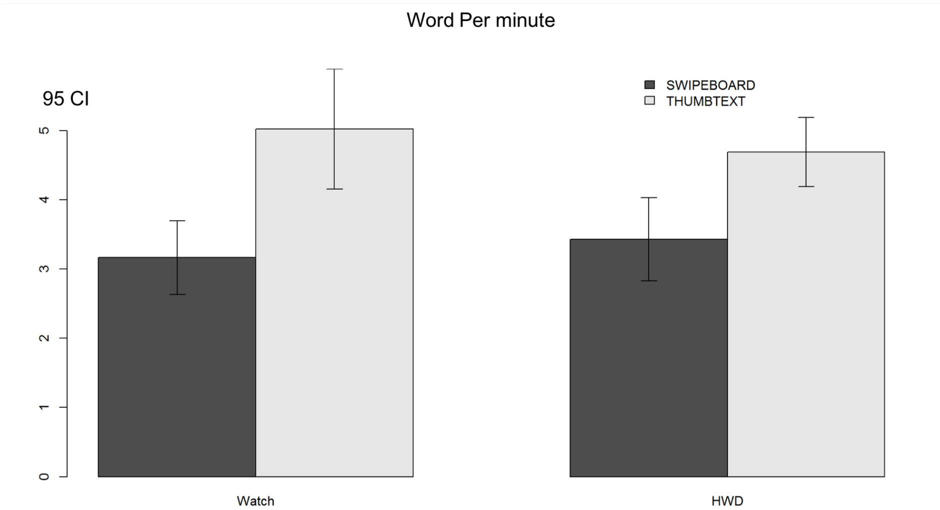


Figure 6.8: WPM Comparing between two devices

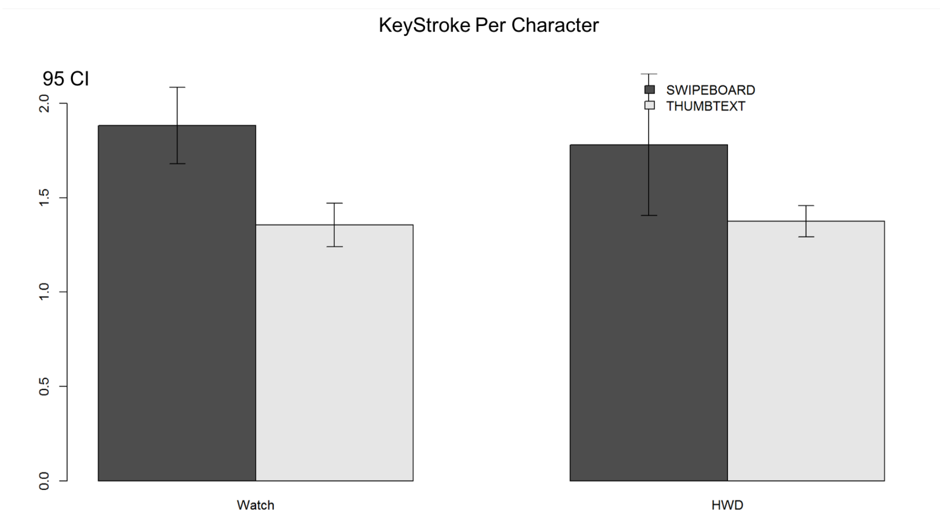


Figure 6.9: KSPC Comparing between two devices

the WPM measure to compare ThumbText and SwipeBoard. Because inputting two 'space' characters lead to early validations with ThumbText and numerous additional characters with SwipeBoard, I only considered trials with a transcribed text length between 60% and 140% of the presented text length. I removed 2.08% of the data (4 trials for SwipeBoard and 8 trials for ThumbText). I performed two-ways ANOVAs and accounted for repeated measures by treating participants as a random variable. I did not find any main effect of *Device* on all metrics nor interaction effect with *Technique*. Thus, I report only results regarding *Techniques* compared with paired t-tests. I discuss only significant differences between ThumbText and SwipeBoard. With ThumbText, participants were faster than with SwipeBoard. This is promising as participants needed to visually search the characters since they did not know the keyboard layout. ThumbText allows for 4.9 WPM. This value is slightly higher than the one we predicted in the previous experiment and could be a result of learning (albeit kept to a minimum in this study).

I also found large error rates for SwipeBoard. I obtained values higher than the one reported in previous studies involving SwipeBoard, i.e. 17.48%. I offer three explanations:

1. *Experimental protocol*: In my experiment, incorrect selections were added to the input sentence without any feedback. In contrast, participants in previous studies received feedback in case of errors [13], or the incorrect character was not added to the input sentence [21]. Thus, my participants had to (and were instructed to) correct their errors without help from the experimental protocol. This hypothesis goes along with the next one,

2. *Diagonal swipe gestures*: It appears that performing a diagonal swipe gesture with the thumb on the ring touchpad was difficult. This is confirmed by qualitative feedback from participants (9/12) who explicitly mentioned the difficulty to enter the

'space' and 'backspace' characters, placed by executing two diagonals consecutively (lower left for Backspace, and lower right for Space). Thus, a correction often resulted in an additional error due to the difficulty to select the 'backspace' key. This lead to a recursive scenario in which participants then had to enter two 'backspace' characters, potentially leading to additional errors.

3. *Direct vs. Indirect input*: SwipeBoard and other similar smartwatch techniques were designed for direct input, i.e. non-distinct visual and input spaces. In my case, the device operates using indirect input. This indirectness, necessitates a 'guessing' phase, to decide the swiping direction. For instance, distinguishing a diagonal swipe vs. a horizontal swipe without sight could mislead users.

Although I adapted SwipeBoard for text-entry on a miniature surface, the use of diagonal swipe gestures is detrimental when using the thumb on a ring touchpad. The ring input is suited at transferring across displays since performance metrics were not affected by the display type.

7

CONCLUSION AND FUTURE WORKS

7.1 THE TECHNIQUE: THUMBTEXT

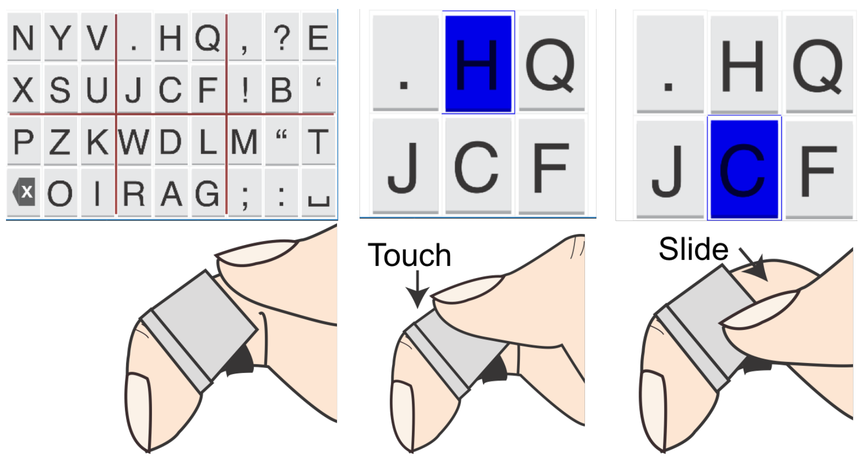


Figure 7.1: Example entering 'c' with the technique

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 [1, 43], disambiguated by a slide-and-lift action [39, 44]. For the ThumbText final design, although I determined that this discretization (2x3) was the most efficient, I also need to determine the effect of the cell position on performances. Cells in the middle column are more error prone than cells on the left and right sides of the ring. This is likely due to the fact that the edge areas can benefit from haptic feedback from the sides of ring.

7.2 DISCUSSION

I first highlight how ThumbText used on a ring touchpad meets my requirements and overcome my challenges. I then discuss how this work and ThumbText fit in the body of work related to text-entry techniques.

7.2.1 *Requirements and Challenges*

I designed ThumbText on a ring touchpad to meet my requirements for a universal text-entry technique on wearable devices. ThumbText uses an indirect input so that it can be used with any wearable display. I showed that ThumbText can transfer between devices without any loss of performances. In addition, I instructed participants to use ThumbText eyes-free during my experiments, ensuring that users could use ThumbText while being mobile. Mobility is also enabled by the fact that the ring can be used with one hand without being actively held. The ring is convenient and always accessible. Small finger inputs on the ring touchpad allow ThumbText to be resistant to environment noises and minimizes the input footprint. The small form factor of both my ring touchpad and current wearable devices introduces challenges for designing a text-entry technique. I reported a multi-step design process through formal studies to assess a target layout (experiment 1 and 2) and an input technique (experiment 2) well suited for the ring form-factor. With ThumbText, novice users can input text with a speed of 4.86 WPM, with a total error rate below 20.15%. I further discuss these metrics in the following section. I aimed at providing a text-entry solution that could be used in combination of numerous wearable displays.

7.2.2 *ThumbText and Previous Work*

I put into perspective performance metrics obtained with ThumbText and other established techniques from the literature. I consider two commonly reported metrics: WPM and KSPC.

7.2.2.1 *WPM*

With ThumbText, novice users have an entry speed of 4.86 WPM (Experiment 4 - Device Effect). This is a value similar or higher than the ones reported in previous work regarding novice performances. For instance, novice users have an entry speed of 2-3 WPM with MDITIM during the first session [27], 3.7 WPM with GesText [28], 5.9 WPM with InclineType [20], 9.09 WPM [13] or 4.5 WPM [21] with SwipeBoard, but only 3.27 WPM with SwipeBoard on my miniature ring touchpad, 7.66 WPM with PalmType [48], 7.6 WPM [41] or 6.0 WPM [32] with Zoomboard, 5.4 WPM with Zshift [32], 7.7 WPM with H4-Writer [36], 5.27 WPM with H4-TEG [5][3], or 6.5 WPM with EdgeWrite [52]. Although not exhaustive, this lists shows that even with challenges of a small ring touchpad, ThumbText offers a text-entry speed (4.86 WPM) in the range of the average (5.62 WPM) plus/minus the standard deviation (1.95 WPM) of these techniques. This is promising as I showed that my ring form factor operated with the thumb dropped the performances of SwipeBoard from 9 WPM to 3.27 WPM. Other text-entry techniques might suffer from the same effect when operated with the thumb on a small input surface.

7.2.2.2 *KSPC*

ThumbText has a KSPC of 1.37, which is less than 2.93 with Quik-Writing [43], 3.05 with MDITIM [27], 1.84 with ZoomBoard [41], 4.3 for EdgeWrite [52], or 1.5 with ZShift [32]. This indicates that the cost of committing errors and fixing them is theoretically lower with Thumb-Text than with other techniques. Interestingly, the performance metrics reported in this work for ThumbText are consistent across wearable devices. This indicates that the indirect input offered by the ring combined with the graphical interface of ThumbText can be used on different device without loss of performance.

7.3 CONCLUSION

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. In this thesis, I introduce ThumbText, a text-entry technique that works across multiple wearable devices to help users from learning multiple device-dependent text-entry techniques. With this technique, 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. I present the results of a multipart design process for the design of ThumbText. 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. I designed ThumbText so that frequent letters are positioned on less error-prone cells. Then, I validated ThumbText and demonstrated that (1) ThumbText offers faster text-entry speed than established text-entry techniques, namely SwipeBoard and H4-Writer in wearable context, and (2) ThumbText can be used across multiple wearable displays without loss of performance.

7.4 FUTURE WORK

In addition to the evaluation of ThumbText with auto-correct and prediction features, there are a few aspects to be examined in the future.

- Longitudinal study: I only performed shorter-version of text entry evaluations. This limits the coverage of characters studied. ThumbText evaluations (experiment 3 and experiment 4) do not cover all letters, only subsets of letters (Experiment 3 covers 20% of English characters' frequency, and experiment 4 covers 32%).

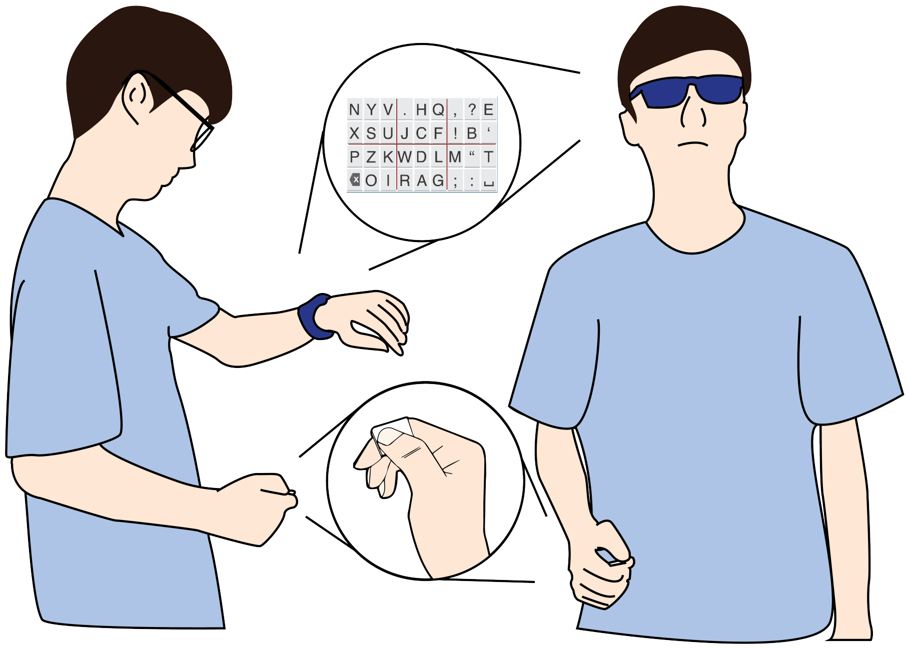


Figure 7.2: Concept image of user entering text with ThumbText

Considering both experiments, I covered over 50% of letter frequency. A longitudinal study can be performed with in-field popular phrase set [35] for at least a few days (multiple sessions) to cover greater than 95% English characters frequency.

- Feature added technique: I was able to determine the users-only performances (8.47 wpm in a difficult context) without any aids from computer,meaning that I did not evaluate the full potential of ThumbText offered by longitudinal study. Results from previous work show maximum of a 82% increase in text-entry rates depending on the presented words [25, 37] that also can be further optimized with auto-correction and/or prediction mechanisms. This indicates that ThumbText could theoretically allow for well over 20 wpm. However, this result remains to be scientifically validated via user experiments.

- Large pool of participants: The sample sizes were not large in experiment 3 (a and b). This can be addressed in the future work by having large pool of participants where the sample size is greater number (e.g. n=15) than the number used in experiment 3 (i.e. n=9) However, the effect sizes, which are independent from sample sizes, were unanimously high according to Cohen's rule of thumb [14]. I also observed *p values* <.05. Therefore, I believe that my data, even with small sample sizes, captured phenomena nicely.
- In-situ study: All the studies in this thesis were conducted in a controlled lab environment. My current prototype is able to communicate via Bluetooth wireless so that the study can be conducted in various environments including realistic scenarios including walking, hand-busy situations. This will evaluate ThumbText's realistic performance.
- Guessability study: I tested two different interaction styles (Tap-tap, and Tap-Slide-Lift). In a guessability study in the future, I can invite potential users to come up with other interaction and interface design with current prototype and also possible integration/improvement of prototype.
- Virtual reality application: Since it offers good opportunity to work with virtually any kind of display, in the future, I can integrate ThumbText device with virtual reality head worn display so that it can be operated within virtual reality environment. It will also validate its usage without looking at the device (i.e. eyes-free to the input device)

A

APPENDIX

A.1 SUPPLEMENT MATERIALS

A.1.1 *Raw visualization of touch input from Experiment 1*

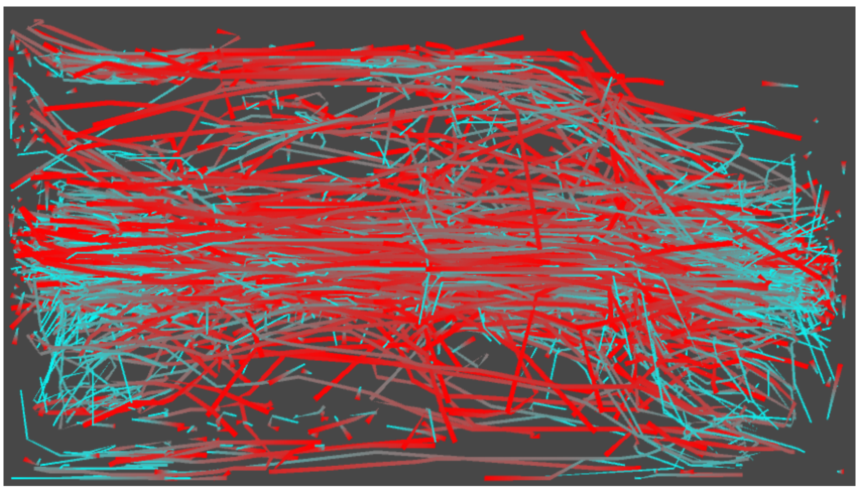


Figure A.1: Visualization of Linear with index finger

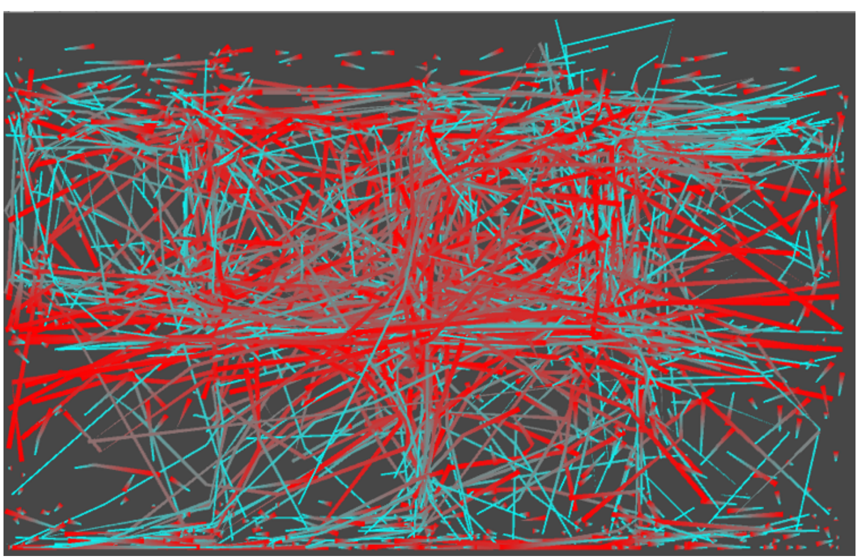


Figure A.2: Visualization of Grid with index finger

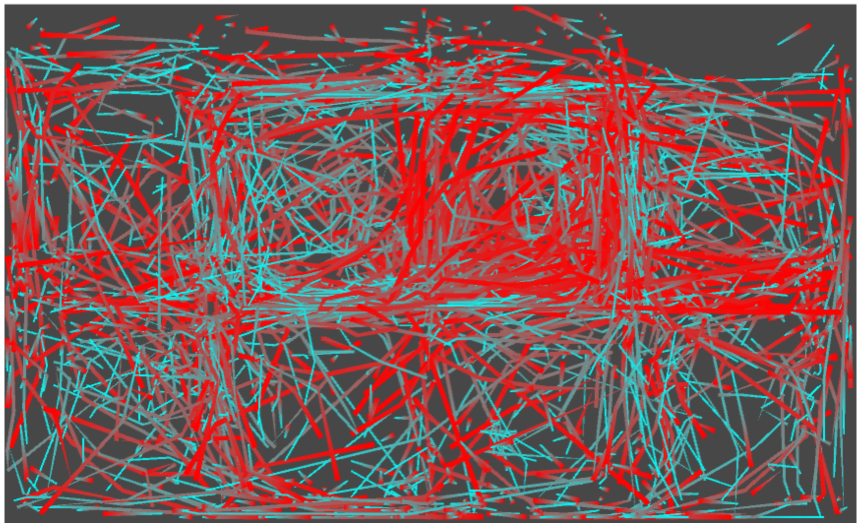


Figure A.3: Visualization of Radial with index finger

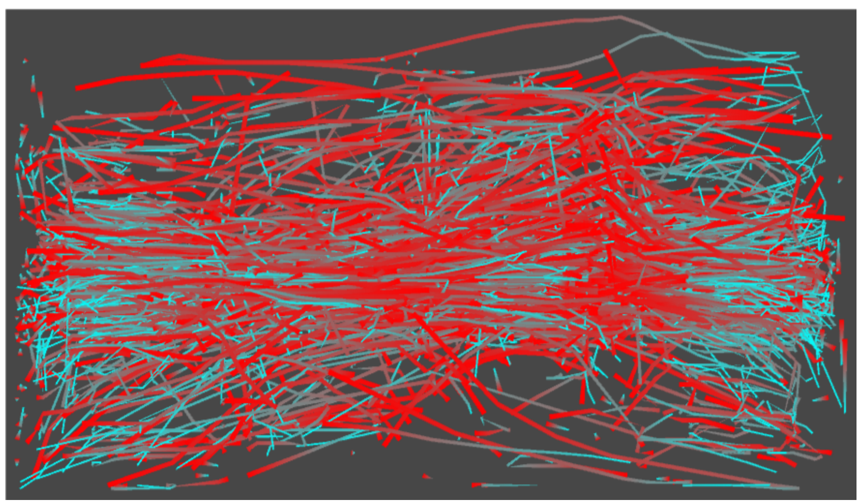


Figure A.4: Visualization of Linear with middle finger

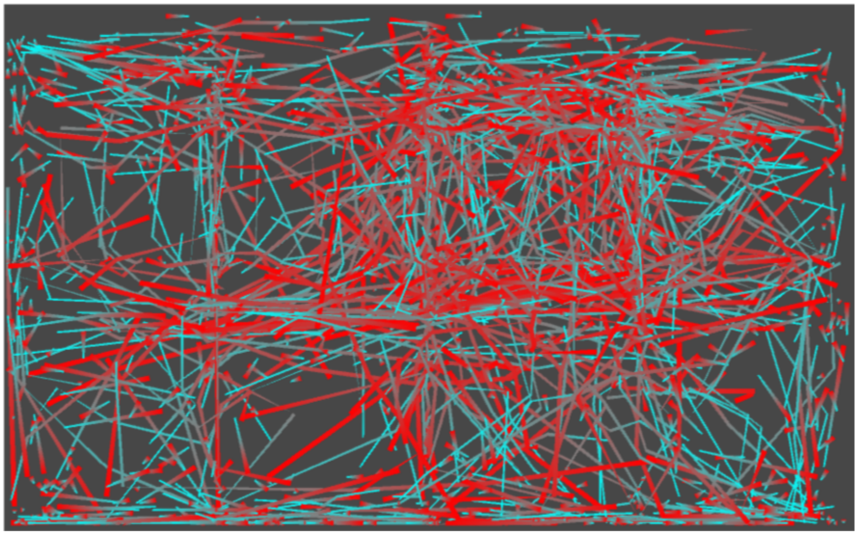


Figure A.5: Visualization of Grid with middle finger

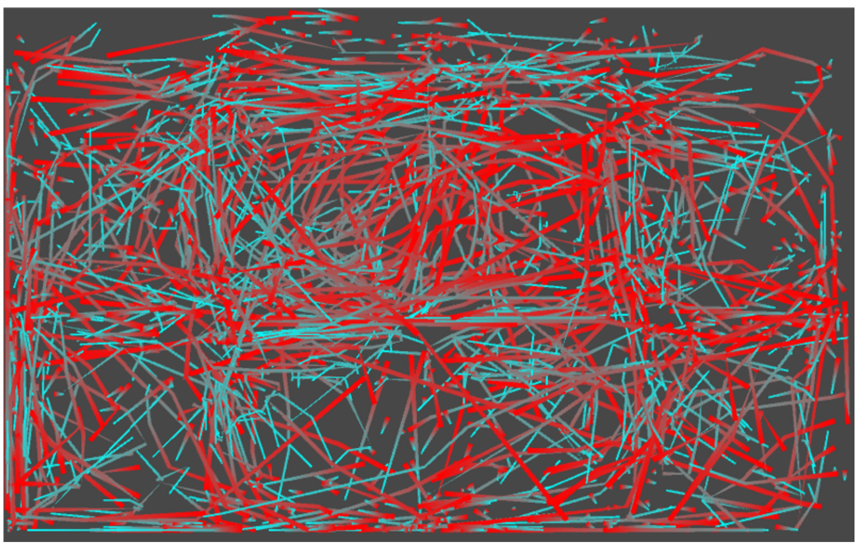


Figure A.6: Visualization of Radial with middle finger

A.1.2 *Set of Words used in Experiments*

Word set for experiment 3a = bill, blip, buff, bulb, bull, fill, flip, flub, fuff, full, luff, lull, pili, pill, puff, pull, pulp

Word set for experiment 3b = adze, bade, bard, bare, bead, bear, brad, bred, dare, daze, dear, drab, raze, read

APPENDIX B

B.1 CONSENT FORM SAMPLE FOR EXPERIMENT

UNIVERSITY OF MANITOBA

Informed Consent Agreement

Please read this consent agreement carefully before you decide to participate in the study.

Purpose of the research study: The purpose of the study is to gain knowledge of human thumb precision on small touchpad when it is in eye-free condition, and to evaluate and compare the precision under different number of conditions.

Research Project Title: TBD

Researchers: Junhyeok Kim (kimj3415@cs.umanitoba.ca)
Dr. William Delamare (William.Delamare@imag.fr)
Supervisor: Professor Dr. Pourang Irani (irani@cs.umanitoba.ca)

Purpose: The purpose of this project is to develop new text entry technique for touch enabled ring type device. This study, as part of the project, is to see how precise human can comfortably interact with given number of cells divided on virtual space without looking at their hand. We use touchpad to capture touch input from participants' thumb and gyroscope on the ring to capture rotational data while operating/interacting with the ring.

This consent form, a copy of which will be left with you for your records and reference, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more details about something mentioned here, or information not included here, you should feel free to ask. Please take the time to read this carefully and to understand any accompanying information.

Participation in this study is voluntary, and will take approximately 1.0 hour of your time. For this study you will be asked to perform tap gestures on touch pad built on top of ring, for coloured target as accurate, quickly as possible on the screen before you. After you complete the tasks you will be asked to complete a questionnaire. All information you provide is considered completely confidential; your name will not be included or in any other way associated, with the data collected in the study. Data collected during this study will be used for academic research and publication purpose in anonymous form. You can also get access to the summary of our findings or the paper, if requested via email. In addition, data will be retained for a period of maximum of one year after publication of results in a locked office in the EITC building, University of Manitoba, to which only researchers associated with this study have access. All data will be destroyed by 07/2017.

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a subject. In no way does this waive your legal rights nor release the researchers, sponsors, or involved institutions from their legal and professional responsibilities. You are free to withdraw from the study at any time, and /or refrain from answering any questions you prefer to omit, without prejudice or consequence. To withdraw, simply inform the researchers above. Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation.

Participant's Signature ________________________ Date ____________

Researcher's Signature ________________________ Date ____________

Figure B.1: A sample informed consent form for participant

BIBLIOGRAPHY

[1] *8Pen*. URL: http://www.8pen.com/.

[2] Gregory D. Abowd and Elizabeth D. Mynatt. «Charting Past, Present, and Future Research in Ubiquitous Computing.» In: *ACM Trans. Comput.-Hum. Interact.* 7.1 (Mar. 2000), pp. 29–58. ISSN: 1073-0516. DOI: 10.1145/344949.344988. URL: http://doi.acm.org/10.1145/344949.344988.

[3] A. S. Arif and W. Stuerzlinger. «Analysis of text entry performance metrics.» In: *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*. 2009, pp. 100–105. DOI: 10.1109/TIC-STH.2009.5444533.

[4] Daniel Ashbrook, Patrick Baudisch, and Sean White. «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*. CHI '11. Vancouver, BC, Canada: ACM, 2011, pp. 2043–2046. ISBN: 978-1-4503-0228-9. DOI: 10.1145/1978942.1979238. URL: http://doi.acm.org/10.1145/1978942.1979238.

[5] B. Bajer, I. S. MacKenzie, and M. Baljko. «Huffman Base-4 Text Entry Glove (H4 TEG).» In: *2012 16th International Symposium on Wearable Computers*. 2012, pp. 41–47. DOI: 10.1109/ISWC.2012.28.

[6] Ravin Balakrishnan and I. Scott MacKenzie. «Performance Differences in the Fingers, Wrist, and Forearm in Computer Input Control.» In: *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. CHI '97. Atlanta, Georgia, USA: ACM, 1997, pp. 303–310. ISBN: 0-89791-802-9. DOI: 10.1145/258549.258764. URL: http://doi.acm.org/10.1145/258549.258764.

[7] Nikola Banovic, Koji Yatani, and Khai N. Truong. «Escape-Keyboard: A Sight-Free One-Handed Text Entry Method for Mobile Touchscreen Devices.» In: *Int. J. Mob. Hum. Comput. Interact.* 5.3 (July 2013), pp. 42–61. ISSN: 1942-390X. DOI: 10.4018/jmhci.2013070103. URL: http://dx.doi.org/10.4018/jmhci.2013070103.

[8] François Bérard and Amélie Rochet-Capellan. «Measuring the Linear and Rotational User Precision in Touch Pointing.» In: *Proceedings of the 2012 ACM International Conference on Interactive Tabletops and Surfaces*. ITS '12. Cambridge, Massachusetts, USA:

ACM, 2012, pp. 183–192. ISBN: 978-1-4503-1209-7. DOI: 10.1145/2396636.2396664. URL: http://doi.acm.org/10.1145/2396636.2396664.

[9] Stuart K. Card, Jock D. Mackinlay, and George G. Robertson. «A Morphological Analysis of the Design Space of Input Devices.» In: *ACM Trans. Inf. Syst.* 9.2 (Apr. 1991), pp. 99–122. ISSN: 1046-8188. DOI: 10.1145/123078.128726. URL: http://doi.acm.org/10.1145/123078.128726.

[10] Steven J. Castellucci and I. Scott MacKenzie. «Graffiti vs. Unistrokes: An Empirical Comparison.» In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy: ACM, 2008, pp. 305–308. ISBN: 978-1-60558-011-1. DOI: 10.1145/1357054.1357106. URL: http://doi.acm.org/10.1145/1357054.1357106.

[11] Jae-Min Cha, Eunjung Choi, and Jihyoun Lim. «Virtual Sliding QWERTY: A new text entry method for smartwatches using Tap-N-Drag.» In: *Applied Ergonomics* 51 (2015), pp. 263 –272. ISSN: 0003-6870. DOI: https://doi.org/10.1016/j.apergo.2015.05.008. URL: http://www.sciencedirect.com/science/article/pii/S0003687015000915.

[12] Liwei Chan, Rong-Hao Liang, Ming-Chang Tsai, Kai-Yin Cheng, Chao-Huai Su, Mike Y. Chen, Wen-Huang Cheng, and Bing-Yu Chen. «FingerPad: Private and Subtle Interaction Using Fingertips.» In: *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. UIST '13. St. Andrews, Scotland, United Kingdom: ACM, 2013, pp. 255–260. ISBN: 978-1-4503-2268-3. DOI: 10.1145/2501988.2502016. URL: http://doi.acm.org/10.1145/2501988.2502016.

[13] Xiang 'Anthony' Chen, Tovi Grossman, and George Fitzmaurice. «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*. UIST '14. Honolulu, Hawaii, USA: ACM, 2014, pp. 615–620. ISBN: 978-1-4503-3069-5. DOI: 10.1145/2642918.2647354. URL: http://doi.acm.org/10.1145/2642918.2647354.

[14] J Cohen. *Statistical power analysis for the behavioral sciences*. 1988. DOI: 10.1234/12345678. arXiv: arXiv:1210.1833v2. URL: http://books.google.com/books?id=Tl0N2lRAO9oC{\&}pgis=1.

[15] B J Davis and J O'Connell. «Shoulder, elbow and wrist components of physiologic tremor amplitudeas measured using a laser penlight.» In: *European neurology* 43.3 (2000), pp. 152–154. ISSN:

0014-3022. DOI: 8155. URL: http://www.ncbi.nlm.nih.gov/pubmed/10765055.

[16] Niklas Elmqvist and Pourang Irani. «Ubiquitous Analytics: Interacting with Big Data Anywhere, Anytime.» In: *Computer* 46.4 (Apr. 2013), pp. 86–89. ISSN: 0018-9162. DOI: 10.1109/MC.2013.147. URL: http://dx.doi.org/10.1109/MC.2013.147.

[17] FlickKey. *FlickKey Keyboard*. 2010. URL: http://www.flickkey.com.

[18] Markus Funk, Alireza Sahami, Niels Henze, and Albrecht Schmidt. «Using a Touch-sensitive Wristband for Text Entry on Smart Watches.» In: *CHI '14 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '14. Toronto, Ontario, Canada: ACM, 2014, pp. 2305–2310. ISBN: 978-1-4503-2474-8. DOI: 10.1145/2559206.2581143. URL: http://doi.acm.org/10.1145/2559206.2581143.

[19] Mitchell Gordon, Tom Ouyang, and Shumin Zhai. «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*. CHI '16. San Jose, California, USA: ACM, 2016, pp. 3817–3821. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858242. URL: http://doi.acm.org/10.1145/2858036.2858242.

[20] Timo Götzelmann and Pere-Pau Vázquez. «InclineType – An Accelerometer-based Typing Approach for Smartwatches.» In: *Proceedings of the XVI International Conference on Human Computer Interaction - Interacción '15*. New York, New York, USA: ACM Press, 2015, pp. 1–4. ISBN: 9781450334631. DOI: 10.1145/2829875.2829929. URL: http://dl.acm.org/citation.cfm?doid=2829875.2829929.

[21] Tovi Grossman, Xiang Anthony Chen, and George Fitzmaurice. «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*. MobileHCI '15. Copenhagen, Denmark: ACM, 2015, pp. 144–152. ISBN: 978-1-4503-3652-9. DOI: 10.1145/2785830.2785867. URL: http://doi.acm.org/10.1145/2785830.2785867.

[22] Aakar Gupta and Ravin Balakrishnan. «DualKey: Miniature Screen Text Entry via Finger Identification.» In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. CHI '16. San Jose, California, USA: ACM, 2016, pp. 59–70. ISBN: 978-1-4503-3362-7. DOI: 10.1145/2858036.2858052. URL: http://doi.acm.org/10.1145/2858036.2858052.

[23] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. «Consumed Endurance: A Metric to Quantify Arm Fatigue of Mid-air Interactions.» In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. Toronto, Ontario, Canada: ACM, 2014, pp. 1063–1072. ISBN: 978-1-4503-2473-1. DOI: 10.1145/2556288.2557130. URL: http://doi.acm.org/10.1145/2556288.2557130.

[24] Jonggi Hong, Seongkook Heo, Poika Isokoski, and Geehyuk Lee. «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*. CHI '15. Seoul, Republic of Korea: ACM, 2015, pp. 1233–1236. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702273. URL: http://doi.acm.org/10.1145/2702123.2702273.

[25] Heidi M. Horstmann and Simon P. Levine. «The effectiveness of word prediction.» In: *Proceedings of the 14th Annual RESNA Conference*. Kansas City: Resna Washington, DC, 1991, pp. 100–102.

[26] Tap Systems Inc. *TAP*. URL: http://www.tapwithus.com.

[27] Poika Isokoski and Roope Raisamo. «Device Independent Text Input: A Rationale and an Example.» In: *Proceedings of the working conference on Advanced visual interfaces - AVI '00*. New York, New York, USA: ACM Press, 2000, pp. 76–83. ISBN: 1581132522. DOI: 10.1145/345513.345262. URL: http://dl.acm.org/citation.cfm?id=345262http://portal.acm.org/citation.cfm?doid=345513.345262.

[28] Eleanor Jones, Jason Alexander, Andreas Andreou, Pourang Irani, and Sriram Subramanian. «GesText: Accelerometer-based Gestural Text-entry Systems.» In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '10. Atlanta, Georgia, USA: ACM, 2010, pp. 2173–2182. ISBN: 978-1-60558-929-9. DOI: 10.1145/1753326.1753655. URL: http://doi.acm.org/10.1145/1753326.1753655.

[29] Amy K. Karlson and Benjamin B. Bederson. «One-handed Touchscreen Input for Legacy Applications.» In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. Florence, Italy: ACM, 2008, pp. 1399–1408. ISBN: 978-1-60558-011-1. DOI: 10.1145/1357054.1357274. URL: http://doi.acm.org/10.1145/1357054.1357274.

[30] Wolf Kienzle and Ken Hinckley. «LightRing: Always-available 2D Input on Any Surface.» In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. UIST '14. Honolulu, Hawaii, USA: ACM, 2014, pp. 157–160. ISBN: 978-1-4503-3069-5. DOI: 10.1145/2642918.2647376. URL: http://doi.acm.org/10.1145/2642918.2647376.

[31] Seoktae Kim, Minjung Sohn, Jinhee Pak, and Woohun Lee. «One-key Keyboard: A Very Small QWERTY Keyboard Supporting Text Entry for Wearable Computing.» In: *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments*. OZCHI '06. Sydney, Australia: ACM, 2006, pp. 305–308. ISBN: 1-59593-545-2. DOI: 10.1145/1228175.1228229. URL: http://doi.acm.org/10.1145/1228175.1228229.

[32] Luis A. Leiva, Alireza Sahami, Alejandro Catala, Niels Henze, and Albrecht Schmidt. «Text Entry on Tiny QWERTY Soft Keyboards.» In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. Seoul, Republic of Korea: ACM, 2015, pp. 669–678. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702388. URL: http://doi.acm.org/10.1145/2702123.2702388.

[33] A R Lingley et al. «A single-pixel wireless contact lens display.» In: *Journal of Micromechanics and Microengineering* 21.12 (2011), p. 125014. URL: http://stacks.iop.org/0960-1317/21/i=12/a=125014.

[34] Kent Lyons, Thad Starner, Daniel Plaisted, James Fusia, Amanda Lyons, Aaron Drew, and E. W. Looney. «Twiddler Typing: One-handed Chording Text Entry for Mobile Phones.» In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '04. Vienna, Austria: ACM, 2004, pp. 671–678. ISBN: 1-58113-702-8. DOI: 10.1145/985692.985777. URL: http://doi.acm.org/10.1145/985692.985777.

[35] I. Scott MacKenzie and R. William Soukoreff. «Phrase Sets for Evaluating Text Entry Techniques.» In: *CHI '03 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '03. Ft. Lauderdale, Florida, USA: ACM, 2003, pp. 754–755. ISBN: 1-58113-637-4. DOI: 10.1145/765891.765971. URL: http://doi.acm.org/10.1145/765891.765971.

[36] I. Scott MacKenzie, R. William Soukoreff, and Joanna Helga. «1 Thumb, 4 Buttons, 20 Words Per Minute: Design and Evaluation of H4-writer.» In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. UIST '11. Santa Barbara,

California, USA: ACM, 2011, pp. 471–480. ISBN: 978-1-4503-0716-1. DOI: 10.1145/2047196.2047258. URL: http://doi.acm.org/10.1145/2047196.2047258.

[37] Tina Magnuson and Sheri Hunnicutt. «Measuring the effectiveness of word prediction: The advantage of long-term use.» In: *TMH-QPSR* 43.1 (2002), pp. 57–67.

[38] Aske Mottelson, Christoffer Larsen, Mikkel Lyderik, Paul Strohmeier, and Jarrod Knibbe. «Invisiboard: Maximizing Display and Input Space with a Full Screen Text Entry Method for Smartwatches.» In: *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '16. Florence, Italy: ACM, 2016, pp. 53–59. ISBN: 978-1-4503-4408-1. DOI: 10.1145/2935334.2935360. URL: http://doi.acm.org/10.1145/2935334.2935360.

[39] Saied B. Nesbat. «A System for Fast, Full-text Entry for Small Electronic Devices.» In: *Proceedings of the 5th International Conference on Multimodal Interfaces*. ICMI '03. Vancouver, British Columbia, Canada: ACM, 2003, pp. 4–11. ISBN: 1-58113-621-8. DOI: 10.1145/958432.958437. URL: http://doi.acm.org/10.1145/958432.958437.

[40] Shahriar Nirjon, Jeremy Gummeson, Dan Gelb, and Kyu-Han Kim. «TypingRing: A Wearable Ring Platform for Text Input.» In: *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. MobiSys '15. Florence, Italy: ACM, 2015, pp. 227–239. ISBN: 978-1-4503-3494-5. DOI: 10.1145/2742647.2742665. URL: http://doi.acm.org/10.1145/2742647.2742665.

[41] Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. «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*. CHI '13. Paris, France: ACM, 2013, pp. 2799–2802. ISBN: 978-1-4503-1899-0. DOI: 10.1145/2470654.2481387. URL: http://doi.acm.org/10.1145/2470654.2481387.

[42] Kurt Partridge, Saurav Chatterjee, Vibha Sazawal, Gaetano Borriello, and Roy Want. «TiltType: Accelerometer-supported Text Entry for Very Small Devices.» In: *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology*. UIST '02. Paris, France: ACM, 2002, pp. 201–204. ISBN: 1-58113-488-6. DOI: 10.1145/571985.572013. URL: http://doi.acm.org/10.1145/571985.572013.

[43] Ken Perlin. «Quikwriting: Continuous Stylus-based Text Entry.» In: *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*. UIST '98. San Francisco, California, USA: ACM, 1998, pp. 215–216. ISBN: 1-58113-034-1. DOI: 10.1145/288392.288613. URL: http://doi.acm.org/10.1145/288392.288613.

[44] Yuan-Fu Shao, Masatoshi Chang-Ogimoto, Reinhard Pointner, Yu-Chih Lin, Chen-Ting Wu, and Mike Chen. «SwipeKey: A Swipe-based Keyboard Design for Smartwatches.» In: *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '16. Florence, Italy: ACM, 2016, pp. 60–71. ISBN: 978-1-4503-4408-1. DOI: 10.1145/2935334.2935336. URL: http://doi.acm.org/10.1145/2935334.2935336.

[45] Tomoki Shibata, Daniel Afergan, Danielle Kong, Beste F. Yuksel, I. Scott MacKenzie, and Robert J.K. Jacob. «DriftBoard: A Panning-Based Text Entry Technique for Ultra-Small Touchscreens.» In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. UIST '16. Tokyo, Japan: ACM, 2016, pp. 575–582. ISBN: 978-1-4503-4189-9. DOI: 10.1145/2984511.2984591. URL: http://doi.acm.org/10.1145/2984511.2984591.

[46] R. William Soukoreff and I. Scott MacKenzie. «Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric.» In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '03. Ft. Lauderdale, Florida, USA: ACM, 2003, pp. 113–120. ISBN: 1-58113-630-7. DOI: 10.1145/642611.642632. URL: http://doi.acm.org/10.1145/642611.642632.

[47] Adam J. Sporka, Torsten Felzer, Sri H. Kurniawan, Ondřej Poláček, Paul Haiduk, and I. Scott MacKenzie. «CHANTI: Predictive Text Entry Using Non-verbal Vocal Input.» In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: ACM, 2011, pp. 2463–2472. ISBN: 978-1-4503-0228-9. DOI: 10.1145/1978942.1979302. URL: http://doi.acm.org/10.1145/1978942.1979302.

[48] Cheng-Yao Wang, Wei-Chen Chu, Po-Tsung Chiu, Min-Chieh Hsiu, Yih-Harn Chiang, and Mike Y. Chen. «PalmType: Using Palms As Keyboards for Smart Glasses.» In: *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*. MobileHCI '15. Copenhagen, Denmark: ACM, 2015, pp. 153–160. ISBN: 978-1-4503-3652-9. DOI: 10.1145/2785830.2785886. URL: http://doi.acm.org/10.1145/2785830.2785886.

[49] Yuntao Wang, Chun Yu, Jie Liu, and Yuanchun Shi. «Understanding Performance of Eyes-free, Absolute Position Control on Touchable Mobile Phones.» In: *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*. MobileHCI '13. Munich, Germany: ACM, 2013, pp. 79–88. ISBN: 978-1-4503-2273-7. DOI: 10.1145/2493190.2493215. URL: http://doi.acm.org/10.1145/2493190.2493215.

[50] Yuntao Wang, Chun Yu, Jie Liu, and Yuanchun Shi. «Understanding Performance of Eyes-free, Absolute Position Control on Touchable Mobile Phones.» In: *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*. MobileHCI '13. Munich, Germany: ACM, 2013, pp. 79–88. ISBN: 978-1-4503-2273-7. DOI: 10.1145/2493190.2493215. URL: http://doi.acm.org/10.1145/2493190.2493215.

[51] Daniel Wigdor and Ravin Balakrishnan. «TiltText: Using Tilt for Text Input to Mobile Phones.» In: *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*. UIST '03. Vancouver, Canada: ACM, 2003, pp. 81–90. ISBN: 1-58113-636-6. DOI: 10.1145/964696.964705. URL: http://doi.acm.org/10.1145/964696.964705.

[52] Jacob O. Wobbrock, Brad A. Myers, and John A. Kembel. «EdgeWrite: A Stylus-based Text Entry Method Designed for High Accuracy and Stability of Motion.» In: *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*. UIST '03. Vancouver, Canada: ACM, 2003, pp. 61–70. ISBN: 1-58113-636-6. DOI: 10.1145/964696.964703. URL: http://doi.acm.org/10.1145/964696.964703.

[53] Jacob Wobbrock. «The future of mobile device research in HCI.» In: (Jan. 2006).

[54] Haijun Xia, Tovi Grossman, and George Fitzmaurice. «NanoStylus: Enhancing Input on Ultra-Small Displays with a Finger-Mounted Stylus.» In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. UIST '15. Charlotte, NC, USA: ACM, 2015, pp. 447–456. ISBN: 978-1-4503-3779-3. DOI: 10.1145/2807442.2807500. URL: http://doi.acm.org/10.1145/2807442.2807500.

[55] Xin Yi, Chun Yu, Weijie Xu, Xiaojun Bi, and Yuanchun Shi. «COMPASS: Rotational Keyboard on Non-Touch Smartwatches.» In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. CHI '17. Denver, Colorado, USA: ACM, 2017, pp. 705–715. ISBN: 978-1-4503-4655-9. DOI: 10.1145/3025453.3025454. URL: http://doi.acm.org/10.1145/3025453.3025454.

[56] Shumin Zhai, Paul Milgram, and William Buxton. «The Influence of Muscle Groups on Performance of Multiple Degree-of-freedom Input.» In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '96. Vancouver, British Columbia, Canada: ACM, 1996, pp. 308–315. ISBN: 0-89791-777-4. DOI: 10.1145/238386.238534. URL: http://doi.acm.org/10.1145/238386.238534.