Investigating a Force-Based Selection Method for Smartwatches in a 1D Fitts' Law Study and Two New Character-Level Keyboards

Yuan Ren
Inclusive Interaction Lab
University of California, Merced
Merced, California, United States
yren5@ucmerced.edu

Ahmed Sabbir Arif
Inclusive Interaction Lab
University of California, Merced
Merced, California, United States
asarif@ucmerced.edu

ABSTRACT

Selecting small targets is difficult on tiny displays due to the "fat-finger problem". In this paper, we explore the possibility of using a force-based approach to target selection on smartwatches. First, we identify the most comfortable range of force on smartwatches. We then conduct a 1D Fitts' law study to compare the performance of tap and force-tap. Results revealed that force-tap is significantly better in selecting smaller targets, while tap outperforms force-tap for bigger targets. We then developed two new force-based keyboards to demonstrate the feasibility of force input in practical scenarios. These single-row alphabetical keyboards enable character-level text entry by performing slides and variating contact force. In a user study, these keyboards yielded about 4 wpm with about 2% error rate, demonstrating the viability of force input on smaller screens.

CCS CONCEPTS

• **Human-centered computing → Text input**; *Gestural input*.

KEYWORDS

out of vocabulary (OOV) word entry, non-ambiguous keyboard, single-row keyboard, force-based input, wearable devices

ACM Reference Format:
Yuan Ren and Ahmed Sabbir Arif. 2023. Investigating a Force-Based Selection Method for Smartwatches in a 1D Fitts' Law Study and Two New Character-Level Keyboards. In *TEI '23: Proceedings of the Seventeenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '23), February 26-March 1, 2023, Warsaw, Poland*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3569009.3572741

1 INTRODUCTION

Smartwatches are becoming increasingly popular among mobile users [22]. However, selecting targets, particularly small targets, is difficult on smartwatches due to the "fat-finger problem" [34]. To facilitate precise target selection, most smartwatch applications either clutter the interface by using large interactive elements or require users to perform a sequence of actions. Both of these approaches affect performance and user preference [20, 28]. In this work, we propose a one-directional force-based target selection approach, with which users slide the finger closer to the target, then variate contact force to move the cursor along the x-axis (reducing the force moves the cursor to the left and increasing the force moves the cursor to the right) to select the target.

Toward this, we first identify the most comfortable range of force on smartwatches. We then compare one-directional force with traditional touch in a Fitts' law experiment. Finally, to demonstrate practical usage of the proposed method, we design and evaluate two new force-based text entry techniques for smartwatches (Section 6). Unlike the existing techniques, these neither occupy most of the display nor use aggressive correction model by disabling character-by-character input. The contribution of this work is thus threefold: identifying the most comfortable range of force on smartwatches, comparison of one-directional force-tap with conventional touch in a Fitts' law study, and the design and evaluation of two new novel force-based keyboards. All studies reported here were approved by the Institutional Review Board (IRB) and conducted abiding by the institute's COVID-19 preventive measures.

TEI '23, February 26-March 1, 2023, Warsaw, Poland

ACM ISBN 978-1-4503-9977-7/23/02.
https://doi.org/10.1145/3569009.3572741

2 RELATED WORK

Not much work focused on target selection on smartwatches. Hara et al. [12] investigated the effects of button size and location on target selection performance with the index finger. They found out 5 mm and 7 mm targets are susceptible to significantly higher selection errors than 10 mm targets. Ishii and Shizuki [18] developed eight different callout features that display and magnify the area occluded by the finger in a non-occluded area. In multiple evaluations, participants found these methods to be useful in target selection. Xia et al. [35] developed a finger-mounted fine-tip stylus to enable fast and accurate pointing with almost no occlusion. In a study, the stylus reduced erroneous selection by 80% compared to traditional touch interaction. Yeo et al. [36] compared target selection in all directions with force-tap, twist, and pan gestures, where force-tap was the most challenging of all methods since it was difficult to apply force in the correct direction. Kurosawa et al. [21] used a tilt and force hybrid method for target selection on a smartwatch. This method uses an electromyography sensor on the arm to detect tilting of the device. To select a target, users first tilt the hand to indicate the cursor direction, then apply force on the arm to move the cursor to the target. Darbar et al. [11] augmented a smartwatch with four pressure sensors to enable users to apply different levels of force on the two sides of the device for zooming, scrolling, and rotating an interactive map. Ahn et al. [1] used a pressure-sensitive wristband to perform similar interactions. These three methods, however, require extramural hardware to function.

(a) The wristband **(b) The custom Apple iOS application**

Figure 1: (a) The wristband used in the study and (b) participants applying force on the screen. The grey, green, and red background colors indicate the initial force level, and correct and incorrect changes in the force level, respectively.

There has been some research on force-based text entry on smartphones. McCallum et al. [25] developed a force-based technique for the standard 12-key mobile keypad by utilizing three levels of force. Likewise, Tang et al. [31] developed a three-key chorded keyboard with three force levels. Both these methods were highly error prone, yielding 9% and 18% error rates in user studies, respectively. Brewster and Hughes [8] presented several pressure-based techniques to switch between uppercase and lowercase letters on a virtual Qwerty, some of which were faster and more accurate than the Shift key. Arif et al. [2], Arif and Stuerzlinger [5] developed an error prevention technique that requires applying extra force on the keys to enter the less probable letters. The effects of the approach on text entry performance were contradictory in two consecutive studies. Arif and Stuerzlinger [6] enabled bypassing auto-correction by applying extra force on the keys. In an evaluation, this approach significantly improved text entry speed and accuracy. Vertanen et al. [32] used a similar approach on smartwatches. Zhong et al. [38] developed a one-dimensional alphabetical keyboard with a sliding cursor over the letters. The cursor covered multiple letters. Users moved the cursor by variating contact force, confirmed selection by performing a quick release (reducing pressure quickly without lifting the finger), the keyboard then disambiguated the input using a probabilistic model. It also enabled entering one character at a time by using a multi-tap [19] like approach. The keyboard displayed the selected letters in descending order of probability, users then multi-tapped on the screen to select the intended letter. In an evaluation, the word- and character-level approaches yielded on average 4 and 11 wpm, respectively, on a smartphone. More recently, Ren and Arif [29] developed a force-based approach to pick numbers from a number wheel on smartwatches. There are, however, no force-based character-level text entry methods available for smartwatches.

3 USER STUDY 1: LEVELS OF FORCE

We conducted a user study to investigate the levels of force users can comfortably apply on smartwatches. The purpose was to map the most comfortable range of force to cursor movements on a tiny display.

3.1 Participants

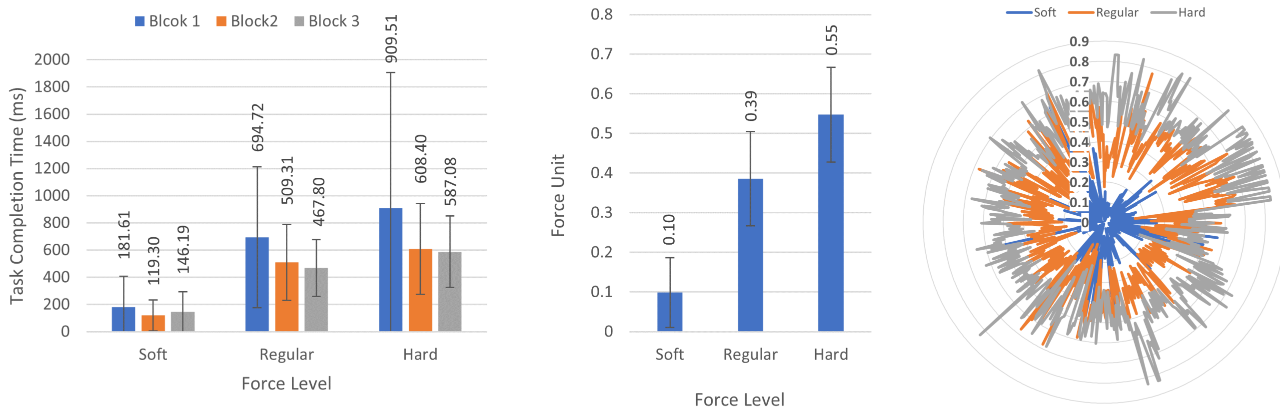
Thirteen participants took part in the study. Their age ranged form 24 to 34 years (M = 28.5, SD = 3.4). Three of them identified themselves as women and ten as men. Nine of them were right-handed and four were left-handed. All of them were experienced mobile users (M = 8.4 years, SD = 2.3). Five of them also owned a smartwatch (M = 0.5 years, SD = 0.9). Six of them had experience with force-based interaction through Apple iOS's 3D touch [7]. They all received U.S. $10 for participating in the study.

3.2 Apparatus

We used an iPhone X (43.6×70.9×7.7 mm, 174 grams) running on iOS version 12.1 at 1125×2436 pixels resolution in the user study. We developed a custom app using the default iOS SDK to simulate an Apple Watch 5's 740 mm^2 display area (312×390 pixels) on the smartphone. We made the surrounding area of the simulated smartwatch touch-insensitive to avoid the effects of accidental touches during the study. We used a smartphone instead of an actual smartwatch since current smartwatches do not provide the support for continuous force detection. Apple Watch detects only the absence and presence of extra force. It is relatively common to use larger devices to study interactions with smartwatches due to technological limitations of current smartwatches [10, 18, 23, 26, 29]. Relevantly, a prior work reported that text entry performances of a keyboard on an actual smartwatch and a simulated smartwatch on a smartphone were comparable in terms of speed and accuracy [37]. To increase the external validity of the work, we replicated not only the interface but also the holding position and posture of a smartwatch. We used a wristband with silicone phone holder (55.5 grams) to attach the smartphone to the wrist of the participants like a smartwatch (Fig. 1a). The wristband held the device on the wrist firmly, thus participants did not have to hold it steady with the fingers of the other hand, although we noticed a few participants occasionally doing that. The holder was 180° rotatable but we did not enable participants to rotate the device during the study to eliminate a potential confound.

3.3 Design & Procedure

First, the participants signed the informed consent form and completed a demographics and mobile usage questionnaire. We then explained the research and demonstrated the custom app and how to variate touch contact force on the screen. Participants were instructed to wear the device on their non-dominant hand and interact using the other hand. All participants were seated, but were instructed not to rest their arms on the desk to increase the external validity of the study. The app displayed one force level on the screen. Participants were instructed to touch the screen,



(a) Task completion time (ms)

(b) Median force per level

(c) Force applied in all tasks

Figure 2: (a) Average task completion time (ms) per block for each force level, (b) median force applied per level, and (c) a radar chart showing force applied in each task. Error bars represent ±1 standard deviation (SD).

then change contact force as instructed on the display, without compromising physical comfort. The area surrounding the simulated smartwatch screen was inactive, but was used to provide feedback on the applied level of force by changing the background color. Initially, the background color was grey, but changed to green when participants changed contact force correctly (reduced force for soft) and to red when changed contact force incorrectly (reduced force for hard) (Fig. 1b). Correct and incorrect force levels were determined based on the increments and decrements in the force value. Specifically, the system identified a correct input when the force value changed in accordance to the displayed force level. For example, for hard force level, the system registered a correct input when the force value was gradually increasing. The system ignored slight variations in the force level (abrupt, discontinuous changes) since it is almost impossible for users to maintain a constant level of force. Once done with one level, the app displayed the next force level. This process continued until all tasks of a block were completed. There were three block, each containing (3 levels × 18 tasks) 54 tasks. Participants were asked to take ∼5 minutes break before starting the next block to mitigate any discomfort due to variating contact force. In summary, the design was: 13 participants × 3 blocks × 3 levels (soft, regular, hard) randomized × 18 tasks = 1,755 data points in total. Upon completion, participants took part in a brief interview discussing their experience in the study.

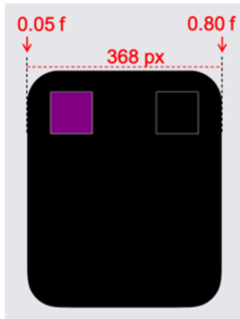
3.4 Results & Discussion

The default Apple iOS SDK returns a value between 0 and 6.67 for the amount of force imparted by the user's finger onto the screen. Similar to Ren and Arif's work [29], we normalized it to the interval from 0 to 1 by dividing the received force value by the maximum force (6.67) for better presentation. The median force applied for soft, regular, and hard tasks were 0.10 (SD = 0.08), 0.39 (SD = 0.12), and 0.55 (0.12), respectively (Fig. 2b). We, thus, decided to use [0.05, 0.80] as the rage of our mapping function, where the lowest value is $\sim \frac{1}{2} \times SD$ from the median of soft force level and the highest value is $\sim 2 \times SD$ from the median of hard force level. These values were picked by closely studying the force patterns of all participants. While participants were fairly consistent in the minimum levels of force applied on the screen, their maximum levels of force varied. In about 10% of all incidents, participants applied a maximum force level closer to 0.8. This suggests that they are comfortable with this force level. Participants also confirmed this in the post-study interview. This encouraged us to increase the maximum value to offer more granularity in the proposed force-based selection method. In Fig. 2c, one can see that force values rarely went outside this range. The values between the range were then mapped to the 368 px horizontal space of the smartwatch using a linear function (Fig. 3). There was a significant effect of level on task completion time ($F_{2,12} = 63.94$, $p < .0001$). On average, soft, regular, and hard tasks took 149.03 ms (SD = 171.2), 557.28 ms (SD = 373.2), and 701.67 ms (SD =643.5) to complete, respectively (Fig. 2a).

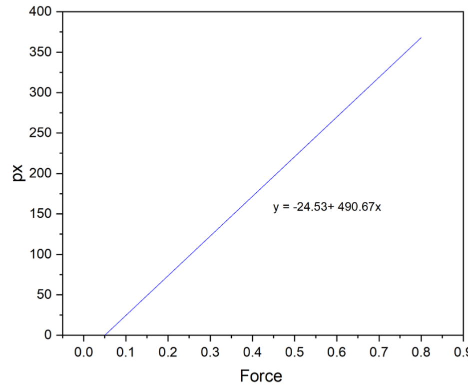
4 1D FITTS' LAW PROTOCOL

Fitts' law is a well-established method for evaluating target selection on computing systems [24]. In the 1990s, it was included in the ISO 9241-9 standard for evaluating non-keyboard input devices by using Fitts' throughput as a dependent variable [30]. Most one-dimensional (1D) Fitts' law experiments combine serial responses with 1D movements. The targets of width W are placed on the two sides of the display (Fig. 4b). The target to select is highlighted. Once selected, the highlight moves to the opposite target. This back-and-forth selection continues until all targets are selected. Each movement covers an amplitude A, which is distance to the centre of the target (Fig. 4a). A *trial* is defined as one target selection task, whereas completing all tasks with a given amplitude is defined as a *sequence*. Throughput cannot be calculated on a single trial because a sequence of trials is the smallest unit of action in ISO 9241-9. Traditionally, the difficulty of each trial is measured in bits using an index of difficulty (ID), calculated as follows:

$$ID = log_2(\frac{A}{W} + 1) \quad (1)$$

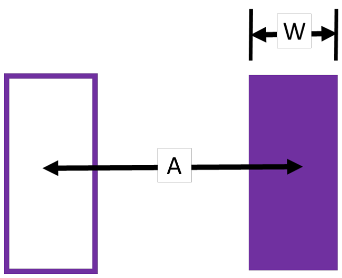


(a) Smartwatch screen

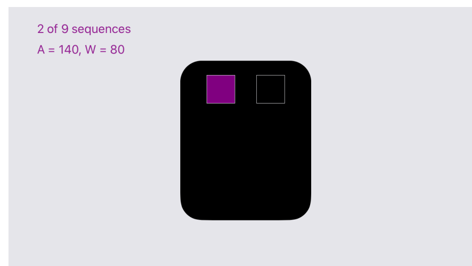


(b) The linear mapping function

Figure 3: The force values within the range [0.05, 0.80] are mapped to the pixels of the display using a linear function.



(a) The 1D Fitts' law task in ISO 9241-9



(b) The custom Apple iOS application

Figure 4: (a) The target is highlighted in purple, (b) the custom Apple iOS app displaying A=140, W=80. It uses the same selection sequence as ISO 9241-9.

The movement time (MT) is measured in seconds for each trial, then averaged over the sequence of trials. It is then used to calculate the performance throughput (TP) in bits/second (bps) using the following equation:

$$TP = \frac{ID}{MT} \tag{2}$$

The revised ISO 9241-9 (9241-411) [17] measures throughput using an effective index of difficult ID_e, which is calculated from the effective amplitude A_e and the effective width W_e to make sure that the real distance traveled form one target to the next is measured. It also takes into account the spread of selections about the target center.

$$TP = \frac{ID_e}{MT} \tag{3}$$

$$ID_e = log_2(\frac{A_e}{W_e} + 1) \tag{4}$$

The effective amplitude is the real distance travelled by the participants, while the effective width is calculated as follows, where SD_x is the standard deviation of the selection coordinates projected on the x-axis for all trials in a sequence. This accounts for any targeting errors by the participants, assuming that participants were aiming at the center of the targets.

$$W_e = 4.133 \times SD_x \tag{5}$$

5 USER STUDY 2: 1D FITTS' LAW STUDY

We conducted a Fitts' law study to compare target selection performance of tap and force-tap. The purpose was to investigate if force-tap could be effective in selecting small targets on smartwatches. We focused only on movements in the x-axis because we envision force as a companion of touch rather than an independent selection method, where users tap in the proximity of a target then move the cursor to the left or right for precise selection. Besides, the difficulties in applying force in all directions is evident in a prior work [36], where force was significantly slower (2,600 ms) and more error prone (1.6%) than twist and pan gestures.

5.1 Participants

Twelve participants took part in the study. Their age ranged from 20 to 34 years (M = 29.3, SD = 2.3). Five of them identified themselves as women and seven as men. Ten of them were right-handed and two were left-handed. They all were experienced mobile device users (M = 10.8 years, SD = 2.7). Nine of them had experience with force-based interaction through Apple iOS's 3D touch [7]. They all received U.S. $15 for participating in the study.

5.2 Apparatus & Design

The study used the same apparatus as the previous study (Section 3.2). A custom app was developed carry out the 1D Fitts' law protocol described above. The experiment was a $2 \times 3 \times 3$ within-subjects design. The independent variables were method (tap, force-tap), amplitude: 80, 140, 200 px (20, 35, 50 mm), and width: 10, 40, 80 px (1.5, 12.5, 20 mm). There were 20 trials per condition. The amplitudes were selected based on the display area, to make sure that the targets do not overlap or go outside the boundary. The widths were selected based on the optimal widths recommended in prior research and design guidelines [12, 13, 15, 16]. The dependent variables were throughput (*TP*) and movement time (*MT*).

5.3 Procedure

The study used the same procedure as the previous study (Section 3.3), except for the tasks, which were in accordance with the 1D Fitts' law protocol discussed in Section 4. In the study, participants selected targets using the two selection methods in a counterbalanced order. The cursor was initially positioned in between the targets, then moved back-and-forth from one target to another by either tap or variating contact force. Participants were instructed to select the targets as fast as possible. Incorrect selections were not allowed, in such cases, participants had to select it again. We enforced a ~5 minute break after a condition to avoid the effect of fatigue. After the completion of both conditions, participants completed the NASA-TLX questionnaire [20] to rate the perceived workload of the methods. They also took part in a brief interview session to discuss their experience in the study.

5.4 Results

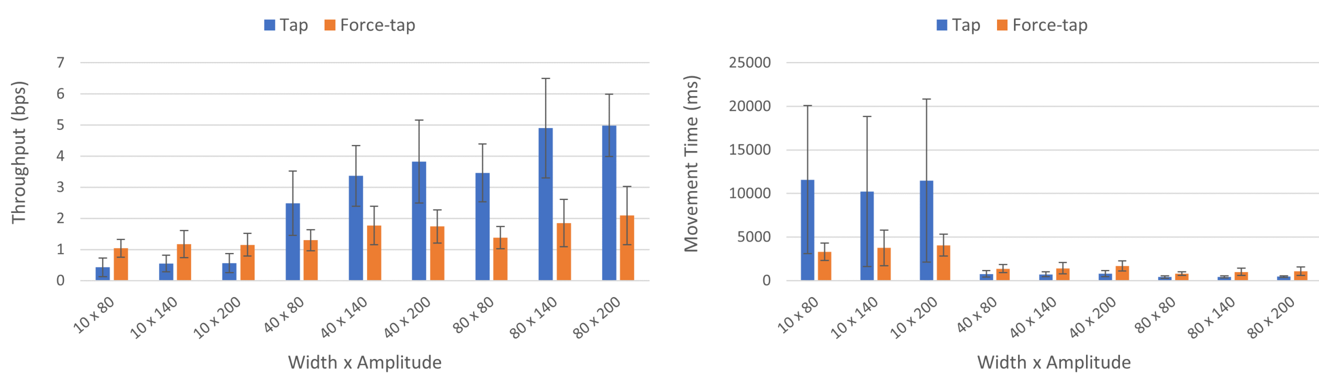
5.4.1 Throughput. An ANOVA identified a significant effect of method on throughput ($F_{1,11} = 76.05, p < .0001$). The average throughput for tap and force-tap were 2.7 bps (SD = 1.8) and 1.5 bps (SD = 0.4), respectively. An ANOVA also identified significant effects of width ($F_{2,11} = 165.50, p < .0001$) and amplitude ($F_{2,11} = 24.56, p < .0001$). The *method* $\times$ *width* interaction effect was also statistically significant ($F_{2,22} = 78.12, p < .0001$). However, the *method* $\times$ *amplitude* interaction effect was not statistically significant ($F_{2,22} = 3.15, p = .06$). Fig. 5a illustrates average throughput for both methods across all examined widths and amplitudes.

5.4.2 Movement Time. An ANOVA identified a significant effect of method on movement time ($F_{1,11} = 5.92, p < .05$). The average movement time for tap and force-tap were 4,098 ms (SD = 5,266) and 2,052 ms (SD = 1,282), respectively. There was also a significant effect of width ($F_{2,11} = 30.78, p < .0001$). However, no significant effect of amplitude was identified ($F_{2,11} = 0.69, p = 0.51$). The *method* $\times$ *width* interaction effect was also statistically significant ($F_{2,22} = 11.02, p < .0005$). But the *method* $\times$ *amplitude* interaction effect was not statistically significant ($F_{2,22} = 0.96, p = 0.4$). Fig. 5b illustrates average movement time for both methods across all examined widths and amplitudes.

5.4.3 Perceived Workload. We present raw TLX scores by analyzing the sub-scales individually, which is a common modification made to NASA-TLX [14]. A Wilcoxon Signed-Rank test failed to identify significant effects of method on mental demand ($z = -0.18, p = .86$), physical demand ($z = -0.1, p = .92$), temporal demand ($z = -0.23, p = .81$), performance ($z = -1.65, p = .10$), and effort ($z = -1.25, p = .21$). However, a significant was identified on frustration ($z = -2.52, p < .05$). Fig. 6c illustrates average NASA-TLX ratings of the two methods.

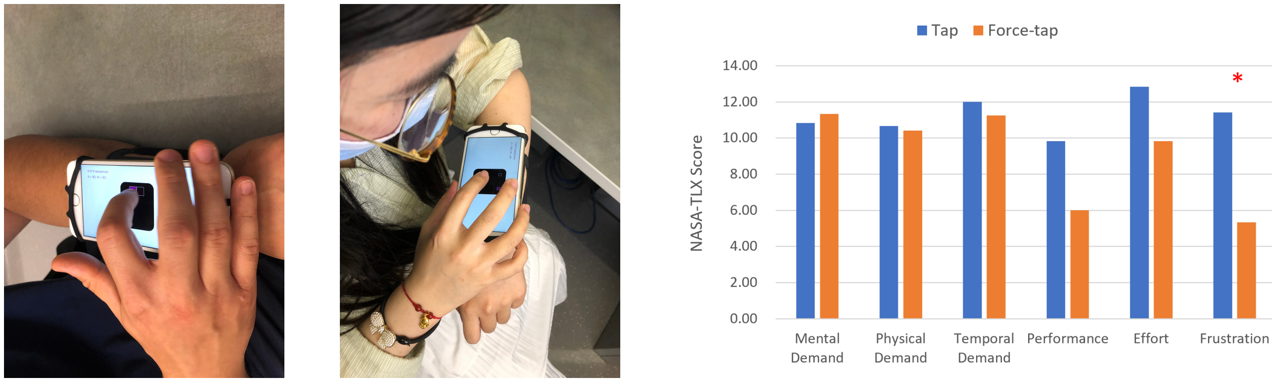
5.5 Discussion

Tap yielded significantly higher throughput than force-tap (~80% higher). However, the interaction effects suggest that throughput of the two methods were significantly affected by the target size and amplitude. A post-hoc Tukey-Kramer test revealed that force-tap performed significantly better with the smallest target (~120% higher throughput), while tap performed significantly better with the bigger ones (~100–151% higher throughput). This is also evident in Fig. 7 that illustrates average throughput for the two selection methods with the three target sizes (10, 40, 80 pixels) fitted to power trendlines. As one can see, both selection methods conformed to



(a) Throughput (bps) **(b) Movement time (ms)**

Figure 5: Average throughput (bps) and movement time (ms) for both methods across all examined widths and amplitudes. Error bars represent ±1 standard deviation (SD).



(a) Tap **(b) Force-tap** **(c) Average NASA-TLX scores**

Figure 6: Two participants selecting targets with (a) tap and (b) force-tap, and (c) average perceived workload of the two methods. The red asterisk indicate statistically significant difference.

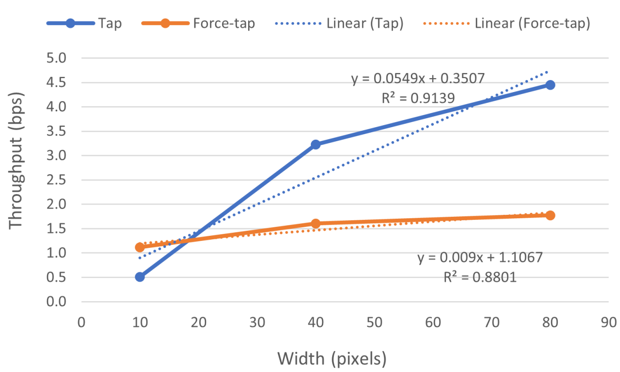


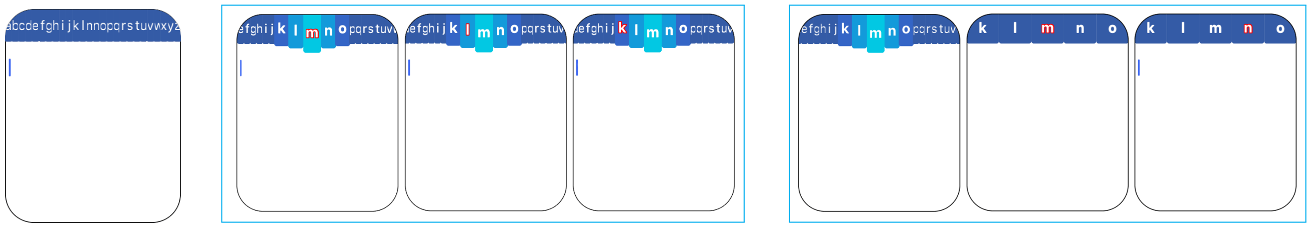
Figure 7: Average throughput (bts) for the two selection methods with the three target sizes fitted to power trendlines.

Fitts' law, as expected — with both, throughput decreased linearly with decreasing target sizes (tap: $R^2 = 0.91$, force-tap: $R^2 = 0.88$), however, tap had a much steeper drop than force-tap.

Overall, force-tap was significantly faster than tap (~50% faster). A deeper analysis indicated that difficulties in selecting the smallest target contributed to this. A post-hoc Tukey-Kramer test revealed that selection time for the smallest target was significantly slower with tap than with force-tap (~67% slower), while selection time for the larger targets were somewhat comparable (see Fig. 5b). These suggest that force could be an effective method for selecting smaller targets on tiny displays. Post-study questionnaire and interview also support this. Participants found both methods relatively similar in terms of mental, physical, and temporal demands, performance, and effort, but were significantly more frustrated with tap than force-tap (see Fig. 6c), primarily due to the smaller targets. One participant (male, 27 years) commented, *"Some blocks have very tiny boxes and it was very hard to hit the right place. It was frustrating."* Another participant (female, 29 years) said, *"Both touch and force seemed the same for bigger squares."* Participants also commented on the learnability of force-tap. One participant (male, 34 years) stated, *"Force is pretty novel to me, it took me a little time to get used to the smallest one. Once I got used to it, I could finish the task faster than beginning."* Some participants, on the other hand, preferred using force-tap exclusively on smartwatches. One participant (female, 27 years) commented, *"It is much easier for me to complete the task of force than touch."* Based on the findings, we recommend enabling both tap and force-tap on smartwatches. Tap is fast and reliable for bigger targets, but for those occasional smaller targets, force-tap is much more reliable, faster, and causes less frustration.

6 SLICE KEYBOARDS

To demonstrate practical usage of force input, we designed two slide and force (slice) keyboards that enable users to enter one character at a time using force-tap and finger slide gestures. We used text entry as our test scenario because entering text is an extreme case of target selection, requiring repetitive selection of different targets (the keys). We designed two alphabetical slice keyboards that display all letters of the English alphabet in a 368 × 70 px row (Fig. 8a). To enter a letter, the user slides her finger horizontally along the x-axis anywhere on the screen. The letter closest to the x-coordinate of the finger and the two neighbouring letters from each side (five in total) are magnified using a zoom-in effect. The slide-force keyboard highlights the central letter (Fig. 8b). The user can reduce contact force to highlight the left letters or increase contact force to highlight the right letters. Realising touch enters the highlighted letter. The slide-force-slide keyboard does not highlight the magnified letters (Fig. 8c), instead requires the user to apply extra force to replace the keyboard with magnified versions of the five letters (73.6 × 70 px each). The initial zoom-in mode of both keyboards are identical. But since the slide-force-slide keyboard enables target selection via both force and slide, the candidate five letters are displayed on the keyboard area to facilitate sliding. Initially, the central letter is highlighted, but the user can side her finger over any key, then release touch to enter the corresponding letter. Both keyboards enable the entry of space and backspace by performing swift left and right strokes, respectively, anywhere on the screen.



(a) Initial state (b) Slide-force keyboard (c) Slide-force-slide keyboard

Figure 8: (a) the initial state of the keyboards showing all letters from 'a' to 'z', (b) the process of entering the letter 'k' with the slide-force keyboard: the user slides her finger anywhere on the screen in relevance to the target, the keyboard magnifies the five nearby keys, because the target is on the left, the user reduces contact force to highlight the letter, then releases touch to enter it, (c) the process of entering the letter 'n' with the slide-force-slide keyboard: the user slides her finger anywhere on the screen in relevance to the target, she applies extra force to replace the keyboard with magnified versions of the five nearby keys, the user then slides her finger over the target letter and releases touch to enter it.

7 USER STUDY 3: SLIDE-FORCE V. SLIDE-FORCE-SLIDE

We conducted a user study to compare the two keyboards. Apart from evaluating the new keyboards, one purpose of the study was to demonstrate that force-based selection method could be used in practical scenarios.

7.1 Participants

Ten participants took part in the study. Their age ranged from 24 to 34 years (M = 28.2, SD = 2.7). None of them participated in the previous studies. Three of them identified themselves as women and seven as men. Eight of them were right-handed and two were left-handed. They all were experienced mobile device users (M = 10 years, SD = 1.3). Six of them also owned a smartwatch (M = 1.6 years, SD = 1.9). None of them had prior experience with force-based interaction. They all received U.S. $15 for participating in the study.

7.2 Apparatus & Design

The study used the same apparatus as the previous studies (Section 3.2). It was a 2×5 within-subjects design. The independent variables were method (slide-force, slide-force-slide) and block. There were five short English phrases [33] per block. The dependent variables were the commonly used words per minute (wpm) and error rate (%) performance metrics in text entry research [4]. In summary, the design was: 10 participants × 2 methods × 5 blocks × 5 phrases = 500 phrases in total.

7.3 Procedure

The study used the same procedure as the first study (Section 3.3), except for the tasks. During the study, participants transcribed short English phrases from a set [33] using the two keyboards in a counterbalanced order. A custom app displayed one random phrase at a time outside the smartwatch area (Fig. 10). Participants were instructed to read and memorize the phrase, then transcribe it as fast and as accurate as possible using either of the method. Error correction was recommended but not forced. Once entered, the app automatically displayed the next phrase. This continued until all phrases were entered. We enforced a ~5 minute break between the conditions to avoid the effect of fatigue. We enabled participants to practice with the methods by entering five phrases with each technique before the corresponding condition. These phrases were not repeated in the main study. After the study, participants completed a custom questionnaire to rate the performance of the methods. They also took part in a brief interview session to discuss their experience in the study.

7.4 Results

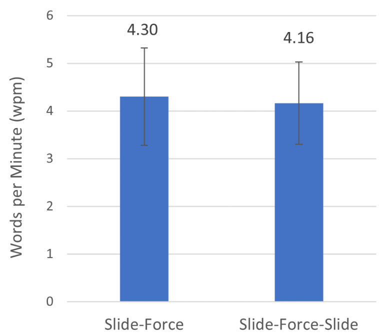
7.4.1 Entry Speed. An ANOVA failed to identify a significant effect of method on entry speed ($F_{1,9} = 0.28, p = .61$). On average slide-force and slide-force-slide yielded 4.3 wpm (SD = 1.0) and 4.2 wpm (SD = 0.7), respectively (Fig. 9a). However, there was a significant effect of block ($F_{4,9} = 30.12, p < .0001$). Fig. 9b illustrates average entry speed of the two methods across all blocks.

7.4.2 Error Rate. An ANOVA failed to identify a significant effect of method on error rate ($F_{1,9} = 0.86, p = .38$). On average SF and SFS yielded 2.2% (SD = 4.5) and 1.7% (SD = 3.7) error rates, respectively (Fig. 9c). There was also no significant effect of block ($F_{4,9} = 1.39, p = .26$).

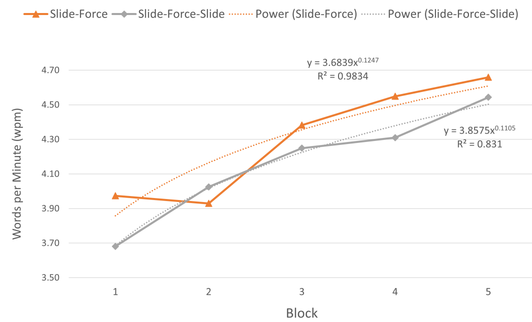
7.4.3 User Feedback. A Wilcoxon Signed-Rank test failed to identify significant effects of method on perceived speed ($z = -0.14, p = .89$), accuracy ($z = -1.26, p = .21$), learnability ($z = -1.29, p = .20$), ease-of-use ($z = -1.0, p = .32$), functionality of the features ($z = 0, p = 1.0$), confidence in using the methods ($z = -0.33, p = .74$), and willingness to use the methods on their smartwatches ($z = -0.21, p = .83$). Fig. 10c illustrates average user ratings of the two methods.

7.5 Discussion

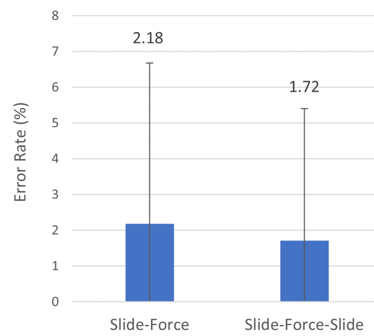
Entry speed of the two methods were comparable. Besides, the 4.3 wpm is much lower than the performance of the existing character-level methods, which were reported to yield between a 4.3 and 19.6 wpm entry speed [3]. These methods, however, used much larger keys by occupying up to 85% of the screen [27] and/or evaluated in longer sessions and on much larger smartwatches. It is also important to note that, we observed a significant effect of block



(a) Entry speed (wpm)



(b) Entry speed (wpm) per block



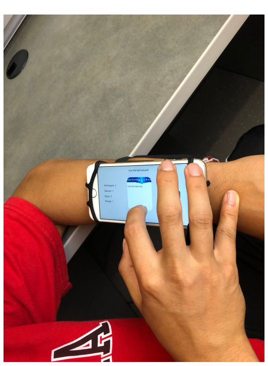
(c) Error rate (%)

Figure 9: (a) Average entry speed (wpm) of the two methods, (b) average entry speed (wpm) across all blocks fitted to power trendlines, (c) average error rate (%) of the two methods. Error bars represent ±1 standard deviation (SD).

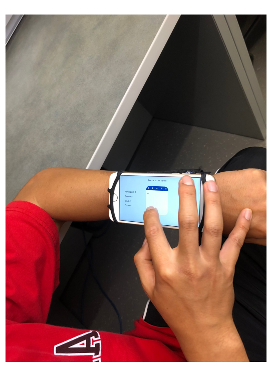
on entry speed. Words per minute with both techniques improved substantially in the last block compared to the first (17% and 23% improvements with slide-force and slide-force-slide, respectively). A post-hoc Tukey-Kramer test identified these differences to be statistically significant. Fig. 9b illustrates average entry speed of the two methods in all blocks fitted to power trendlines, where one can see that both slide-force ($R^2 = 0.98$) and slide-force-slide ($R^2 = 0.83$) correlated well with the power law of practice [9]. These suggest that learning occurred with both methods even in the short duration of the study, thus possible that performance with the methods will improve further with practice. Many participants also felt that their performance improved with practice. One participant (male, 30 years) commented, *"It took some time to get to used to it, but after that it was easy to use."* There was no significant effect of method on error rate. Both methods were fairly accurate with about 2% error rate, which is much lower than the 5–28% error rate reported for the existing character-level methods for smartwatches [3]. Participants were mostly indifferent about the two methods in the post-study questionnaire (Fig. 10c). However, we noticed that participants were split about which method they preferred the most. One participant (female, 29 years) commented, *"The split-force-split was the fastest for me to use even through I hadn't used it before."*, while another (male, 34 years) said, *"To sum up, slide-force is the best way to type for me."* However, many participants found both methods difficult to learn. One participant (female, 24 years) commented, *"They [both] are quite hard to control how much force need to push when choose the letters."* These results suggest that while the keyboards may not be appropriate as the primary method of text entry on smartwatches, these could be used as extensions to the primary input method (which are predominantly predictive with aggressive correction model, thus do not always enable out-of-vocabulary word entry [27]) for non-dictionary word entry or to enter short phrases (e.g., short response to a text message). Most importantly, the fact that force input prevailed even in an extreme scenario like text entry indicate that it can be effectively used an active mode of interaction on smaller devices.

8 CONCLUSION

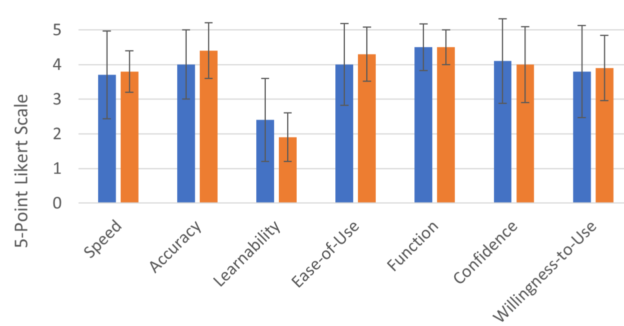
In this work, we investigated the possibility of using contact force as an active mode of interaction on smartwatches, especially to enable the selection of smaller targets. We presented the results of three user studies. The first identified the most comfortable range



(a) Slide-Force



(b) Slide-Force-Slide



(c) Average user ratings

Figure 10: A participant entering text with (a) slide-force and (b) slide-force-slide keyboards, and (c) average user ratings of the two methods on a 5-point Likert scale (1–5: low–high). Error bars represent ±1 standard deviation (SD).

of force users can apply on smartwatches. The second revealed that force input is significantly more effective in selecting smaller targets than touch. Finally, the third study demonstrated that force could be effectively used in practical scenarios by developing and comparing two new force-based character-level text entry techniques. In addition to the means for demonstrating force input's effectiveness, we see these keyboards as independent contributions as they have much smaller footprints than the existing character-level methods and users were relatively fast at learning these. We envision these keyboards being used as extensions to predictive keyboards that disable out-of-vocabulary word entry due to their aggressive correction models, to enable the entry of occasional non-dictionary words.

9 FUTURE WORK

One limitation of the work is the studies reported here were conducted on simulated smartwatch interfaces on a smartphone. While this is fairly common in the literature [10, 18, 23, 26, 29], due to the absence of empirical evidence, it is unclear if the performance recorded on a simulated smartwatch is generalizable to actual smartwatches. In the future, we will explore different control-display mapping functions for force input. We will also use machine learning approaches to make the method more reliable. We also hope to evaluate the new keyboards in longitudinal studies and augment them with predictive systems for faster text entry.

REFERENCES

[1] Youngseok Ahn, Sungjae Hwang, HyunGook Yoon, Junghyeon Gim, and Junghee Ryu. 2015. BandSense: Pressure-Sensitive Multi-Touch Interaction on a Wristband. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15)*. ACM, New York, NY, USA, 251–254. https://doi.org/10.1145/2702613.2725441 event-place: Seoul, Republic of Korea.

[2] Ahmed Sabbir Arif, Mauricio Herrera López, and Wolfgang Stuerzlinger. 2010. Two New Mobile Touchscreen Text Entry Techniques. In *Poster at the 36th Graphics Interface Conference (GI '10)*. CEUR-WS.org/Vol-588, Toronto, ON, Canada, 22–23.

[3] Ahmed Sabbir Arif and Ali Mazalek. 2016. A Survey of Text Entry Techniques for Smartwatches. In *Human-Computer Interaction. Interaction Platforms and Techniques*, Masaaki Kurosu (Ed.). Vol. 9732. Springer International Publishing, Cham, 255–267. https://doi.org/10.1007/978-3-319-39516-6_24 Series Title: Lecture Notes in Computer Science.

[4] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2009. Analysis of Text Entry Performance Metrics. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*. IEEE, Washington, DC, USA, 100–105. https://doi.org/10.1109/TIC-STH.2009.5444533

[5] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2013. Evaluation of a New Error Prevention Technique for Mobile Touchscreen Text Entry. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration (OzCHI '13)*. Association for Computing Machinery, New York, NY, USA, 397–400. https://doi.org/10.1145/2541016.2541063

[6] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2013. Pseudo-Pressure Detection and Its Use in Predictive Text Entry on Touchscreens. In *Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collaboration (OzCHI '13)*. ACM, New York, NY, USA, 383–392. https://doi.org/10.1145/2541016.2541024 event-place: Adelaide, Australia.

[7] Dieter Bohn. 2015. Here's what 3D Touch can do on the iPhone 6S. https://www.theverge.com/2015/9/22/9370739/3d-touch-features-iphone-6s

[8] Stephen A. Brewster and Michael Hughes. 2009. Pressure-Based Text Entry for Mobile Devices. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '09)*. ACM, New York, NY, USA, 9:1–9:4. https://doi.org/10.1145/1613858.1613870 event-place: Bonn, Germany.

[9] Stuart K. Card, Allen Newell, and Thomas P. Moran. 2000. *The Psychology of Human-Computer Interaction*. L. Erlbaum Associates Inc., USA.

[10] Xiang 'Anthony' Chen, Tovi Grossman, and George Fitzmaurice. 2014. Swipeboard: A Text Entry Technique for Ultra-Small Interfaces That Supports Novice to Expert Transitions. In *Proceedings of the 27th Annual Acm Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 615–620. https://doi.org/10.1145/2642918.2647354

[11] Rajkumar Darbar, Prasanta Kr Sen, and Debasis Samanta. 2016. Presstact: Side Pressure-Based Input for Smartwatch Interaction. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. ACM, New York, NY, USA, 2431–2438. https://doi.org/10.1145/2851581.2892436 event-place: San Jose, California, USA.

[12] Kiyotaka Hara, Takeshi Umezawa, and Noritaka Osawa. 2015. Effect of Button Size and Location When Pointing with Index Finger on Smartwatch. In *Human-Computer Interaction: Interaction Technologies (Lecture Notes in Computer Science)*, Masaaki Kurosu (Ed.). Springer International Publishing, Cham, 165–174. https://doi.org/10.1007/978-3-319-20916-6_16

[13] Aurora Harley. 2019. Touch Targets on Touchscreens. https://www.nngroup.com/articles/touch-target-size/

[14] Sandra G. Hart. 2006. Nasa-Task Load Index (NASA-TLX); 20 Years Later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50, 9 (Oct. 2006), 904–908. https://doi.org/10.1177/154193120605000909 Publisher: SAGE Publications Inc.

[15] Android Accessibility Help. 2022. Touch Target Size - Android Accessibility Help. https://support.google.com/accessibility/android/answer/7101858?hl=en

[16] Apple Inc. 2022. Buttons - Menus and actions - Components - Human Interface Guidelines - Design - Apple Developer. https://developer.apple.com/design/human-interface-guidelines/components/menus-and-actions/buttons/

[17] International Organization for Standardization. 2012. ISO/TS 9241-411:2012. https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/05/41/54106.html

[18] Akira Ishii and Buntarou Shizuki. 2016. Exploring Callout Design in Selection Task for Ultra-Small Touch Screen Devices. In *Proceedings of the 28th Australian Conference on Computer-Human Interaction (OzCHI '16)*. Association for Computing Machinery, New York, NY, USA, 426–434. https://doi.org/10.1145/3010915.3010922

[19] Christina L. James and Kelly M. Reischel. 2001. Text Input for Mobile Devices: Comparing Model Prediction to Actual Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '01)*. Association for Computing Machinery, New York, NY, USA, 365–371. https://doi.org/10.1145/365024.365300

[20] Tomonari Kamba, Shawn A. Elson, Terry Harpold, Tim Stamper, and Piyawadee Sukaviriya. 1996. Using Small Screen Space More Efficiently. In *Proceedings of the SIGCHI conference on Human factors in computing systems common ground - CHI '96*. ACM Press, Vancouver, British Columbia, Canada, 383–390. https://doi.org/10.1145/238386.238582

[21] Hiroki Kurosawa, Daisuke Sakamoto, and Tetsuo Ono. 2018. MyoTilt: A Target Selection Method for Smartwatches Using the Tilting Operation and Electromyography. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '18)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3229434.3229457

[22] Paul Lamkin. 2018. Smart Wearables Market to Double by 2022: $27 Billion Industry Forecast. https://www.forbes.com/sites/paullamkin/2018/10/23/smart-wearables-market-to-double-by-2022-27-billion-industry-forecast/?sh=2a1c63442656 Link to the article.

[23] Luis A. Leiva, Alireza Sahami, Alejandro Catala, Niels Henze, and Albrecht Schmidt. 2015. Text Entry on Tiny QWERTY Soft Keyboards. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 669–678. https://doi.org/10.1145/2702123.2702388

[24] I. Scott MacKenzie. 2018. Fitts' Law. In *The Wiley Handbook of Human Computer Interaction*. John Wiley & Sons, Ltd, Hoboken, NJ, USA, 347–370. https://doi.org/10.1002/9781118976005.ch17

[25] David C. McCallum, Edward Mak, Pourang Irani, and Sriram Subramanian. 2009. PressureText: Pressure Input for Mobile Phone Text Entry. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09)*. Association for Computing Machinery, New York, NY, USA, 4519–4524. https://doi.org/10.1145/1520340.1520693

[26] Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. 2013. ZoomBoard: A Diminutive Qwerty Soft Keyboard Using Iterative Zooming for Ultra-Small Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2799–2802. https://doi.org/10.1145/2470654.2481387 event-place: Paris, France.

[27] Gulnar Rakhmetulla and Ahmed Sabbir Arif. 2021. SwipeRing: Gesture Typing on Smartwatches Using a Segmented QWERTY Around the Bezel. In *Proceedings of Graphics Interface 2021 (GI 2021)*. Canadian Information Processing Society, Toronto, ON, Canada, 166 – 177. https://doi.org/10.20380/GI2021.19 ISSN: 0713-5424 event-place: Virtual Event.

[28] Gulnar Rakhmetulla, Ahmed Sabbir Arif, Steven J. Castellucci, I. Scott MacKenzie, and Caitlyn E. Seim. 2021. Using Action-Level Metrics to Report the Performance of Multi-Step Keyboards. In *Proceedings of Graphics Interface 2021 (GI 2021)*.

Canadian Information Processing Society, Toronto, ON, Canada, 127 – 137. https://doi.org/10.20380/GI2021.15 ISSN: 0713-5424 event-place: Virtual Event.

[29] Yuan Ren and Ahmed Sabbir Arif. 2021. Stepper, Swipe, Tilt, Force: Comparative Evaluation of Four Number Pickers for Smartwatches. *Proceedings of the ACM on Human-Computer Interaction* 5, ISS (Nov. 2021), 500:1–500:21. https://doi.org/10.1145/3488545

[30] R. William Soukoreff and I. Scott MacKenzie. 2004. Towards a Standard for Pointing Device Evaluation, Perspectives on 27 Years of Fitts' Law Research in HCI. *International Journal of Human-Computer Studies* 61, 6 (Dec. 2004), 751–789. https://doi.org/10.1016/j.ijhcs.2004.09.001

[31] HUI Tang, DAVID J. Beebe, and ARTHUR F. Kramer. 2001. A Multilevel Input System with Force-Sensitive Elements. *International Journal of Human-Computer Studies* 54, 4 (April 2001), 495–507. https://doi.org/10.1006/ijhc.2000.0451

[32] Keith Vertanen, Dylan Gaines, Crystal Fletcher, Alex M. Stanage, Robbie Watling, and Per Ola Kristensson. 2019. VelociWatch: Designing and Evaluating a Virtual Keyboard for the Input of Challenging Text. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300821 Paper 591.

[33] Keith Vertanen and Per Ola Kristensson. 2011. A Versatile Dataset for Text Entry Evaluations Based on Genuine Mobile Emails. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. Association for Computing Machinery, New York, NY, USA, 295–298. https://doi.org/10.1145/2037373.2037418

[34] Daniel Vogel and Patrick Baudisch. 2007. Shift: A Technique for Operating Pen-Based Interfaces Using Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. ACM, New York, NY, USA, 657–666. https://doi.org/10.1145/1240624.1240727 event-place: San Jose, California, USA.

[35] Haijun Xia, Tovi Grossman, and George Fitzmaurice. 2015. 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)*. Association for Computing Machinery, New York, NY, USA, 447–456. https://doi.org/10.1145/2807442.2807500

[36] Hui-Shyong Yeo, Juyoung Lee, Andrea Bianchi, and Aaron Quigley. 2016. WatchMI: Pressure Touch, Twist and Pan Gesture Input on Unmodified Smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '16)*. ACM, New York, NY, USA, 394–399. https://doi.org/10.1145/2935334.2935375 event-place: Florence, Italy.

[37] Xin Yi, Chun Yu, Weinan Shi, and Yuanchun Shi. 2017. Is It Too Small?: Investigating the Performances and Preferences of Users When Typing on Tiny Qwerty Keyboards. *International Journal of Human-Computer Studies* 106 (Oct. 2017), 44–62. https://doi.org/10.1016/j.ijhcs.2017.05.001

[38] Mingyuan Zhong, Chun Yu, Qian Wang, Xuhai Xu, and Yuanchun Shi. 2018. ForceBoard: Subtle Text Entry Leveraging Pressure. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–10. https://doi.org/10.1145/3173574.3174102