# Stepper, Swipe, Tilt, Force: Comparative Evaluation of Four Number Pickers for Smartwatches

YUAN REN and AHMED SABBIR ARIF, Human-Computer Interaction Group, University of California, Merced, United States


Fig. 1. The four number pickers investigated in this work, from left: input stepper (the default number picker on most smartwatches), swipe (in this figure, the user is swiping down to decrease a value), tilt (in this figure, the user is tilting up to increase a value), and force (in this figure, the user is variating contact force to change a value). Unlike the default input stepper, the new methods do not display numeric values in full-screen when selected, instead display a magnifier window to display the value changing (see the last image).

500

Picking numbers is arguably the most frequently performed input task on smartwatches. This paper presents three new methods for picking numbers on smartwatches by performing directional swipes, twisting the wrist, and varying contact force on the screen. Unlike the default number picker, the proposed methods enable users to actively switch between slow-and-steady and fast-and-continuous increments and decrements during the input process. We evaluated these methods in two user studies. The first compared the new methods with the default input stepper method in both stationary and mobile settings. The second compared them for individual numeric values and values embedded in text. In both studies, swipe yielded a significantly faster input rate. Participants also found the method faster, more accurate, and the least mentally and physically demanding compared to the other methods. Accuracy rates were comparable between the methods.

CCS Concepts: • **Human-centered computing** → **Gestural input**; *Graphical user interfaces*; *Empirical studies in interaction design.*

Additional Key Words and Phrases: input and interaction, numbers, editing, pressure, touchscreen

**ACM Reference Format:**
Yuan Ren and Ahmed Sabbir Arif. 2021. Stepper, Swipe, Tilt, Force: Comparative Evaluation of Four Number Pickers for Smartwatches. *Proc. ACM Hum.-Comput. Interact.* 5, ISS, Article 500 (November 2021), 21 pages. https://doi.org/10.1145/3488545

Authors' address: Yuan Ren, yren5@ucmerced.edu; Ahmed Sabbir Arif, asarif@ucmerced.edu, Human-Computer Interaction Group, University of California, Merced, 5200 N. Lake Road, Merced, California, United States, 95343.


2573-0142/2021/11-ART500. https://doi.org/10.1145/3488545

## 1 INTRODUCTION

Smartwatches are becoming increasingly popular among mobile users. A recent survey reported that roughly one-in-five U.S. adults (21%) regularly wear a smartwatch or a fitness tracker [66]. Yet, interaction with these devices is mostly limited to receiving notifications on emails, text messages, and social media activities [13]. This is primarily due to the smaller screen sizes of smartwatches, which limits the types of actions users can perform on these devices. Text entry on these devices is particularly difficult as most existing text entry methods use miniature keys that are difficult to select, causing frequent errors [6]. Some methods also use multi-step approaches, where users have to perform a sequence of actions to input one character. Due to these challenges in text entry, number picking is arguably the most frequent input task performed on smartwatches. For illustrative purposes, current Apple smartwatches come with 40 pre-installed apps[1], of which four partially support alphanumeric text and emoticon entry, eight support number entry using a number picker, and the remaining 28 are only for acquiring and viewing information.

Unlike traditional input methods, with which users enter one digit of a number at a time (such as a virtual keyboard), number pickers display a default value on the screen and enable users to pick a different number either from a list or by increasing or decreasing the default value using a two-segment control. The default number picker on most smartwatches is "input stepper". It uses a spinning number wheel metaphor, where flicking up on the screen spins the wheel to increase the value and spinning the wheel by a down flick decreases the value. Faster flicks spin the wheel faster and vice versa. Once flicked, the wheel keeps on spinning for some time, gradually slowing down to a full stop. Thus, repeated flicks are needed for continuous spinning of the wheel. Input stepper is also the default number picker on most smartphones. However, on smartphones, this method is used by a select number of apps (i.e., the clock) since the dominant method for entering numeric and other characters on these devices is virtual keyboards, while on smartwatches it is used almost exclusively to enter numbers. Section 3.1 describes this method. There are several limitations of this method. First, it does not usually enable editing inline values (numbers embedded in text, e.g., *"12:30"* in *"let's meet at 12:30 pm"*). Even when it does, it displays the spinner in full-screen, forcing users to take their attention away from the details on the screen that may include important details about the target number (e.g., available time slots for a meeting). Second, this method is impractical when the difference between the default and the intended values is very large (e.g., when changing *"\$50 daily"* to *"\$1,500 monthly"*) as it requires repetitive flicks, which can be physically and mentally challenging. Yet, to the best of our knowledge, no prior works studied number pickers on smartwatches or proposed alternatives to the default number picker.

In this paper, we develop three new number pickers in a rigorous design process. We then compare the performance of these methods with the default method in two empirical studies: one comparing their performance in stationary and mobile settings and another with individual and inline numeric values. The remainder of the paper is organized as follows. First, we review the existing works in the area. We then describe the default number picker and introduce the proposed number pickers. We discuss the design considerations for the new methods. We then present the findings of two user studies comparing the four number pickers. Finally, we conclude by reflecting on future extensions of the work.

## 2 RELATED WORK

Numerous works have explored swipe-based, tilt-based, and force-based input and interaction methods for larger touchscreen-based devices like interactive tabletops and walls [37, 56], tablets [30, 67], and smartphones [2, 15, 17, 22, 23, 59, 61–63]. However, these alternative interaction

[1] Apps on Apple Watch, https://support.apple.com/en-gb/guide/watch/apdf1ebf8704/watchos

methods have not been well explored in the context of smartwatches. Interaction with smartwatches is fundamentally different from these devices as not only they are smaller in size but also have different holding and usage patterns. Unlike other touchscreen-based devices, smartwatches are worn on the wrist, which limits its interaction space as users cannot use the fingers of the watch-hand to interact with the device.

There has been an increased interest in text entry methods for smartwatches. Majority of these methods are miniature versions of the standard Qwerty layout that enable users to increase the size of the keys by either tapping or flicking on the screen [18, 38, 57]. There are also alternative miniature layouts that map the English alphabet to a fewer number of keys than Qwerty, then disambiguate the input using a language model [34, 42]. Two recent methods, WatchWriter [32] and SwipeRing [60], enable users to enter text by connecting the keys or the zones containing the letters of the intended word on the screen. Another method, WrisText [31], enables connecting the zones by whirling the wrist of the watch-hand in joystick-like motions. However, these methods use language models to disambiguate the input, and do not offer effective mechanisms for entering numeric values. Arif and Mazalek [6] provide a comprehensive review of the existing text entry methods for smartwatches.

Some have used smartwatches to control other systems. Duet enables controlling a smartphone using the spatial configuration of a smartwatch [19]. Users can tap and swipe on the smartwatch and perform wrist gestures to interact with a smartphone. Likewise, MultiFi lets users use extended widgets on a smartwatch in augmented/mixed reality, where users select a menu item by either tapping on the screen or pointing at the item. Some have also used smartwatches as active tangible in tangible-tabletop systems [33]. Actible [29], for example, augments a smartwatch with custom hardware to enable an expanded set of tangible interactions on interactive tabletops, including shaking, tilting, stacking, neighboring, and on-screen gestures [5, 25, 51]. These methods, however, were designed keeping interaction with other systems in mind, thus are not suitable for interacting with the smartwatches. Besides, the smartwatch-based active tangibles are usually used on a tabletop, not worn on the wrist. A different line of research exploits the accelerometer and gyroscope sensors of smartwatches [68, 69] or external sensors (e.g., infrared sensors [41, 43] and chest-mounted camera [49]) to track hand and finger movements. These works, however, are outside the scope of this research.

A few works have also explored tilt-based and force-based interaction methods on smartwatches. Dunlop et al. [27] propose a semi-transparent interface for switching between a full-screen Qwerty and a full-screen text input area by tilting the wrist. An evaluation found the method to be more error-prone than a conventional method. Kurosawa et al. [45] use 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. A similar work [24] augments a smartwatch with four pressure sensors: two on the left and two on the right side of the watch. It enables 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], in contrast, use a pressure-sensitive wristband to perform similar interactions. These methods require extramural hardware to function. Besides, the former method yielded a much slower task completion time compared to a conventional method in a user study, while the latter two were not compared with the state-of-the-art.

In a different work, Mo and Zhou [54] investigated the interrelationship between smartwatch interaction methods (tapping, swiping, and wrist flicking) and users' movements (standing, strolling, walking, rushing, and jogging) and gait features. Results revealed that both smaller target sizes and interaction on the move decrease the effectiveness and efficiency of tapping. Besides, tapping, swiping, or wrist flicking on the go reduces users' gait symmetry and step length. A similar study

[40] investigated the fatigue caused by different poses, such as sitting with the arm rested, standing with the arm raised, etc., when selecting targets using pointing, dwelling, and swipes. Results showed that the standing position leads to a massive increase in exertion than sitting.

To the best of our knowledge, no prior work developed or compared swipe-based, tilt-based, and force-based interaction methods on a smartwatch, especially in the context of number entry.

Table 1. Actions associated with the four number pickers.

| Method | Increment | Decrement | Acceleration Rate | Continuous Spinning |
|---|---|---|---|---|
| Input Stepper | Flick up | Flick down | Pace of flick | Repetitive flicks |
| Swipe-Based | Swipe up | Swipe down | Length of swipe | Swipe and hold |
| Tilt-Based | Tilt the wrist up | Tilt the wrist down | Angle of tilt | Tilt and hold |
| Force-Based | Increase contact force | Decrease contact force | Level of force | Hold contact force |

## 3 THE FOUR NUMBER PICKERS

This section presents the default input stepper and the new swipe-based, tilt-based, and force-based number pickers. The design of the new methods were refined in iterative design steps. Table 1 summarizes the functionality of the four methods. In addition, these methods share the following behaviors.

- **Selection.** All methods enable selecting a numeric value for increment or decrement by tapping on it. When the selected value is embedded in text (inline values), the input stepper displays a full-screen virtual number wheel containing all legal values, which is the default behavior on most smartwatches. The swipe-based, tilt-based, and force-based methods, in contrast, do not display the number wheel in full-screen, instead change the value directly in the text. For conjoint values (i.e., multiple numbers connected with infixes), the input stepper displays one number wheel per segment. For example, for the time value "12:30:44", it displays three wheels, one for each segment, which is the default behavior on most smartwatches. With the new methods, however, users individually tap on the three parts of the value to change the respective parts.
- **Continuous Spinning.** The new methods enable continuous spinning of the number wheel with a single action for faster increment and decrement when the difference between the current and the intended value is large (e.g., changing "110" to "250"). The default input stepper does not provide the support for this, rather requires users to repeatedly flick on the screen for continuous spinning of the wheel. This feature is further discussed in Sections 3.1.
- **Auditory Feedback.** All four methods provide auditory feedback on spinning the number wheel (a spinning wheel sound, like the default Apple iOS input stepper).
- **Visual Feedback.** Since the new methods do not switch to a full-screen mode when users select a numeric value embedded in text, they provide additional visual feedback to assure that users can see the value changing when their finger occludes the embedded value. Particularly, these methods display a magnifier window further from the finger (Fig. 1).

For the design and development, we simulated an Apple Watch 5 on an Apple iPhone X, where only the smartwatch display was active (Section 4.1 provides further details). We could not use an actual smartwatch since current smartwatches do not provide the support for continuous force detection. Apple Watch 5 includes a force sensor that can only *"distinguish between a light tap and a deep press"* [28]. We optimized all methods for the simulated smartwatch for a fair comparison between them. It is relatively common to use smartphones or tablets to study interactions with smartwatches due to technological limitations of current smartwatches (e.g., [18, 46, 57]).

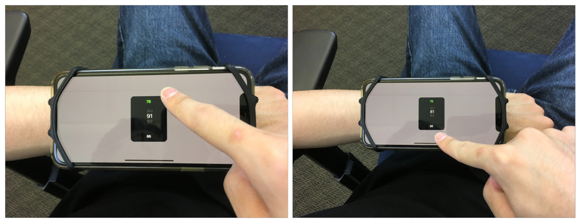


Fig. 2. Input stepper is the default number picker on most smartwatches: it displays a full-screen number wheel that users spin by flicking up or down to increment or decrement a numeric value, respectively.

### 3.1 Input Stepper

Input stepper is the default number picker on most smartwatches (Fig. 2). However, different mobile operating systems use different names to refer to this method, such as *Spinner* and *Spinning wheel*. When users tap on a numeric value, it displays a full-screen virtual number wheel containing all legal values. Spinning the wheel by flicking up on the screen increases the value and spinning the wheel by flicking down decreases the value[2]. The pace of the increment and decrement is determined based on how fast users flick on the screen. Faster flicks rotate the wheel faster and slower flicks rotate the wheel slower. When the desired number is displayed, users pick it by tapping on it. Continuous spinning is not fully supported by the default method. Once flicked, the wheel keeps on spinning for some time, gradually slowing down to a full stop. Hence, repeated flicks are needed for continuous spinning of the wheel. We implemented this method using the default iOS SDK control library[3], without any customization.

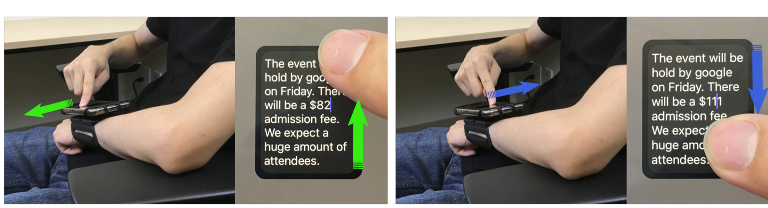



Fig. 3. Swipe-based number picker: the user swipes up to increment and swipes down to decrement a numeric value.

### 3.2 Swipe-Based Picker

The swipe-based picker enables users to pick numbers by performing swipes (Fig. 3). Tapping on a numeric value activates an invisible number wheel containing all legal values. Users then swipe up or down on the screen to increment or decrement the value, respectively. One difference between

[2]Some smartwatches use a reversed mapping, where spinning the wheel by flicking up decreases the value and spinning the wheel by flicking down increases the value. We decided against using this mapping to maintain interaction consistency between the four examined number pickers.
[3]Apple Developer, Pickers - Controls - iOS - Human Interface Guidelines, https://developer.apple.com/design/human-interface-guidelines/ios/controls/pickers.

the default input stepper and swipe is: with the former, users perform a short quick flick, while with the latter, users perform a steady swipe. A single swipe changes the value by one unit. Users can activate continuous spinning of the number wheel by stroking and holding the swipe for 850 milliseconds. Lifting the finger deactivates continuous spinning. The pace of a spin is determined based on the length of the swipe. Holding a *long* swipe increases or decreases the value faster, while holding a *short* swipe increases or decreases the value slowly. This enables users to actively pick a change rate appropriate for the task. Based on the findings of a pilot study ($N$ = 12, M = 26.0 years) investigating various custom and commonly used functions [14, 20] for mapping control movements to the movements of a display object, referred to as the control-display (CD) gain, we used the following function to determine the pace of a spin.

$$t(l) = \frac{1}{2} \times \left(1 - \frac{l}{h}\right)^3 \tag{1}$$

Where $t(l)$ is the pace of a spin relative to the length of a swipe $l$ in pixels, and $h$ is the height of the active smartwatch touchscreen in pixels. Note that we also considered a temporal approach to determine the pace of a spin, where the pace increases in proportion to the duration of a swipe-hold (the wheel keeps on spinning faster as users continue with holding a swipe). However, it performed poorly compared to the proposed approach in terms of speed and accuracy in another pilot study ($N$ = 6, M = 26.0 years). Besides, most participants found the mapping confusing.

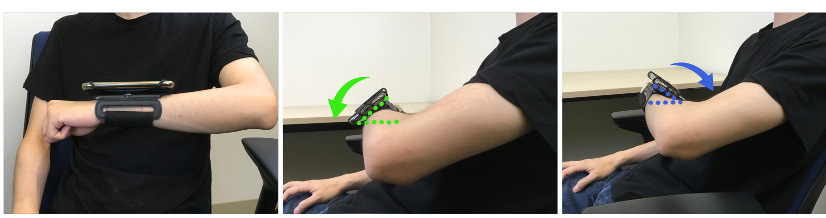


Fig. 4. Tilt-based number picker: the user tilts the device away from the body (up) or towards the body (up) to increment and decrement a numeric value, respectively.

### 3.3 Tilt-Based Picker

The tilt-based method enables users to pick numbers by tilting their smartwatch towards or away from their body, interpreted as tilting down and up, respectively (Fig. 4). Tilting the device once then returning to the initial position changes the value by one unit. For continuous spinning of the invisible number wheel, users tilt and hold the position for 850 milliseconds. Bringing the device back to its initial position deactivates continuous spinning. The pace of a spin is determined based on the angle of the tilt. Tilting the device at a steeper angle increases or decreases the value faster, while tilting it at a slight angle increases or decreases the value slowly. This enables users to actively pick a change rate appropriate for the task. First, we conducted a pilot study ($N$ = 6, M = 26.0 years) to explore various custom and commonly used functions for CD gain [14, 20] to map different tilt angles to different spin pace. Selecting the most effective function was particularly challenging since tilting the device too much not only caused irritation and fatigue but also made the text illegible. Hence, based on Dunlop et al's [27] recommendation, we tested 15° ± 9 angle as a possible range for tilt. However, it was proven to be too sensitive for precise pace section in a second pilot study ($N$ = 12, M = 27.08 years). Besides, we observed that tilting the device towards the body is more difficult than tilting away from the body, thus using the same range for the up and down tilts is impractical. Based on this, we changed the range and selected the following function to determine the pace of a spin based on the findings of a third pilot study ($N$ = 12, M = 26.01 years).

$$t(a) = \begin{cases} \frac{3}{5} \times \left(1 - \frac{|a - up\_optimal|}{up\_threshold}\right)^{3.2} & \textit{when tilting up} \\ \frac{3}{5} \times \left(1 - \frac{|a - down\_optimal|}{down\_threshold}\right)^{3} & \textit{when tilting down} \end{cases} \tag{2}$$

Where $t(a)$ is the pace of a spin relative to the tilt angle $a$, $up\_threshold$ and $down\_threshold$ are the maximum possible up and down tilts $[-40°, 30°]$ based on the anatomy of the human wrist [48], and $up\_optimal$ and $down\_optimal$ are the optimal up and down tilts $[-30°, 20°]$ based on the findings of the pilot studies. Like the swipe-based method, we also tested a temporal approach to determine the pace of a spin, where holding a tilted position for longer increased the pace of the spin and vice versa. However, it was significantly slower, more error prone, and caused irritation and fatigue in the pilot study.


Fig. 5. Force-based number picker: the user applies extra force to increment and soft force to decrement a numeric value.

### 3.4 Force-Based Picker

The force-based method enables users to pick numbers by variating contact force on the screen. Increasing contact force increases the value and decreasing contact force decreases the value (Fig. 5). 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. We normalized it to the interval from 0 to 1 by dividing the received force value by the maximum force (6.67). Then, based on the findings of a pilot study ($N$ = 15, M = 27.7 years), we segmented this range into three force levels: soft (from 0 to 0.15), regular (from 0.15 to 0.3), and hard (from 0.3 to 1). Initially, we attempted an adaptive approach that adjusted the levels of force based on users' typical contact force. However, this method failed to accurately predict the three levels of force in a pilot study investigating different positions and posture, including when standing, sitting, and walking ($N$ = 15, M = 23.9 years). It also bothered the participants that they could not actively control the force levels. Further, it required a substantial amount of training data for each participant, which can be challenging and often impractical in real-world scenarios. The fixed range approach used in this work, in contrast, performed well in prior studies with smartphones [7, 8], as well as in a third pilot study ($N$ = 13, M = 28.46 years). Participants were able to learn the three levels and replicate those in various settings without any major difficulties.

With the force-based method, changing the level of force once then returning to the regular force changes the value by one unit. For continuous spinning of the invisible number wheel, users change the level of force and hold that level for 850 milliseconds. Changing the level or lifting the finger deactivates continuous spinning. This method uses a temporal approach to determine the pace of a spin in continuous spinning. Holding a specific level of force longer increases or decreases the value faster at the following rates: 1 digit per 600 milliseconds for 2 seconds, 1 digit per 250 milliseconds for 5 seconds, and 1 digit per 5 milliseconds. This incremental pace rate was selected based on the findings of a fourth pilot study ($N$ = 6, M = 26.5 years), where users could alter the contact force to restart the rate, enabling them to actively pick a rate appropriate for a task. To keep users informed, this method provides haptic feedback on each force level change. That is, the device vibrates for 200 milliseconds, as recommended by Kaaresoja and Linjama [39]. Since users do not necessarily have to lift their finger to select a value with this method, once selected, they could slide their fingers to a different value to edit it.

## 4 EXPERIMENT 1: SEATED VS. WALKING

We conducted a user study to compare the four number pickers in both stationary and mobile settings. The study protocol was reviewed and approved by the Institutional Review Board (IRB). The study was completed before the World Health Organization (WHO) declared the outbreak of COVID-19 a pandemic.

### 4.1 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 existing smartwatches cannot detect the exact level of force applied on the screen (see Section 3). Apple Watch detects only the absence and presence of extra force. We used a wristband with silicone phone holder (55.5 grams) to attach the smartphone to the wrist of the participants like a smartwatch (Figure 6). 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. We used a Fitness Reality TRE5000 electric treadmill to simulate walking, which is common practice in controlled studies (e.g., [9, 11, 52, 64]).

Fig. 6. The wristband and the device used in the user study.

### 4.2 Participants

Twelve participants voluntarily took part in the study. None of them participated in the pilot studies. Their age ranged from 20 to 39 years (M = 25.27, SD = 5.5). Six of them identified themselves as female and six identified as male. Ten of them were right-handed and two were left-handed. All right-handed participants chose to wear the device on their left hand and interact with the device using the right hand, while all left-handed participants chose to wear it on the right hand and use the other hand for interaction. They all were experienced smartphone users (M = 8.36 years of experience, SD = 1.9). Two of them owned a smartwatch (M = 3.0 years of ownership, SD = 2.8). Four of them also had experience using force as an input modality on Apple iPhone devices (M = 4.6 years of experience, SD = 3.6). They all received U.S. $10 for participating in the study.

### 4.3 Design

We used a within-subjects design for the study, where the independent variables were *setting* and *method*, and the dependent variables were the following performance metrics.

- **Task completion time** (seconds) is the average time it took to change one presented value to the target value.
- **Actions per task** is the average number of actions, including taps, swipes, tilts, and different levels of force, performed to change one presented value to the target value.

Besides, participants were asked to complete the following questionnaires.

- A usability questionnaire that asked participants to rate various aspects of the examined number pickers on a 7-point Likert scale. It included four questions from the SUS questionnaire [12] and two custom questions. The four questions from SUS were about the willingness to use when seated and walking (SUS Q#1), ease of use (SUS Q#3), and learnability of the methods (SUS Q#7). The two custom questions were about the perceived speed and accuracy of the methods. These questions were used since SUS does not include questions about system speed or accuracy.
- A perceived workload questionnaire that included three questions from the NASA-TLX questionnaire [55] about mental demand (NASA-TLX Q#1), physical demand (NASA-TLX Q#6), and frustration (NASA-TLX Q#6).

We did not use the full SUS and NASA-TLX questionnaires to reduce the time and effort needed in the study. These questionnaires include 16 questions in total, which would have resulted in $(2 \times 4 \times 16 =)$ 128 questions in the study. Hence, we only used the questions that are most relevant to our investigation. Hart [35], the creator of NASA-TLX, identified using a subset of the questions as one of the most common usage of the questionnaire and did not discourage it. Leaving some questions out of the SUS questionnaire is also common [47]. Note that we evaluated each scale individually rather than calculating a single score per questionnaire to eliminate the possibility of biases in factor analyses.

Participants were asked to complete both questionnaires upon the completion of the study to increase the reliability of the data as it enabled them to compare the efforts needed with each method while rating them. We acknowledge that this increases the chance of the *context effect*, which suggests participants tend to rate the latter methods more demanding than the ones they did earlier. However, Hart [36] found it to be *"typical of subjective ratings in general"* and argued that it can be avoided by being *"careful to control context effects"*, which we did by counterbalancing the conditions using a balanced Latin square. This assured that participants experienced the methods in different orders. Prior studies showed it to be an effective approach to mitigate context or order effect [50, pp. 177–181]. In summary, the design was as follows.

12 participants ×
2 settings (seated and walking, *counterbalanced*) ×
4 methods (the default stepper, swipe, tilt, and force, *counterbalanced*) ×
15 random two-digit values between 9 and 100
= 1,440 numeric values in total, excluding practices.

### 4.4 Experimental Tasks

During the study, the app presented one numeric value at a time and asked the participants to change it to a target value using the method under investigation (Fig. 7). Both the presented and the target values were two-digit numbers between 9 and 100. All tasks were randomly generated for each participant, making sure that there were equal number of increment and decrement tasks, and each task had its equal counterpart. That is, the system paired each increment task with an equivalent decrement task and vice versa, where the presented and target values had the same edit distance. For example, for a decrement task: change "68" to "43", there was an equivalent increment task: change "23" to "48" ($68 - 43 = 48 - 23 = 25$). We used numbers between 9 and 100 based on our observation that two-digit numbers, e.g., time (h:m:s), volume (0–100%), etc., are the most commonly edited numeric values on smartwatches[4]. Further, it allowed us to constrain each study session within one hour, reducing any potential effects of fatigue. It also enabled us to evaluate the new methods with numeric values with which the default input stepper is the most effective. The default input stepper method is likely to take more time and effort to edit larger numbers (> 100) as it would require repeatedly flicking on the screen to keep the wheel spinning. While with the proposed methods, users can continue spinning the wheel without performing additional actions until the target number is reached.

### 4.5 Walking Speed and Safety

The treadmill was set on 1.0 mph (~1.6 km/h) during the walking condition. We selected this rate based on a prior study that showed that users usually maintain a walking speed between 0.9 and 1.2 mph (1.5 and 2.0 km/h) when using a mobile phone [53]. Appropriate safety measures were taken during this condition. All participants were asked to attach the treadmill safety key to their clothing and wear a bike helmet to prevent injuries in case of an unexpected slip, trip, or fall. Besides, there were mandatory breaks between the conditions to prevent exhaustion for using the treadmill.

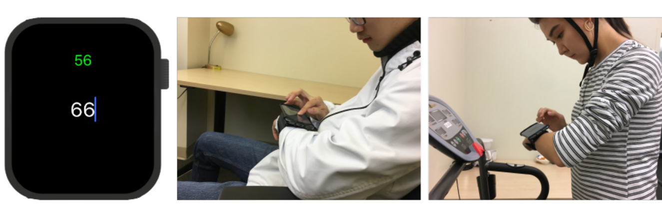



Fig. 7. A decrement task (change "66" to "56") displayed on the study app (left), and two participants taking part in the study while seated and while walking on a treadmill, respectively (right).

### 4.6 Procedure

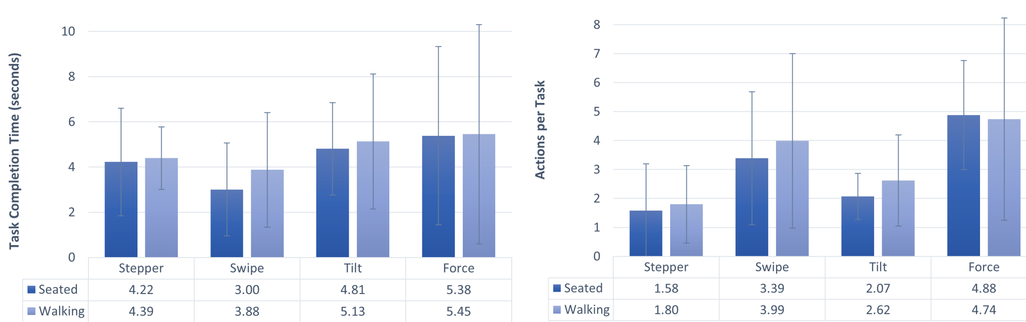
The study was conducted in a quiet room, one participant at a time. Upon arrival, we explained the research to all participants and collected their consents. They then completed a demographics and mobile usage questionnaire. The main study started after that. First, we demonstrated the first method and enabled them to practice with it by performing two increment and two decrement tasks. They were then asked to perform the experimental tasks (Section 4.4) both when seated and when walking in a counterbalanced order. They were instructed to perform the tasks as fast as possible. After successfully completing a task, they tapped on a button outside the smartwatch

[4]Besides, informal investigations suggest that single- and two-digit numbers are the most frequently used [26] and the most popular [10].

area to see the next task. Once done with all tasks, we demonstrated the second method, enabled them to practice with it, and asked them to perform the experimental tasks. This process continued until they experienced all methods. The methods were also introduced in a counterbalanced order. Upon completion of the study, the participants completed a questionnaire where they rated various aspects of the four methods on a 7-point Likert scale, and the perceived mental and physical demands and frustration using the NASA-TLX [55] questionnaire.

### 4.7 Results

A Shapiro-Wilk test revealed that the response variable residuals are normally distributed. A Mauchly's test indicated that the variances of populations are equal. Thus, we used a repeated-measures ANOVA to analyze the quantitative data. In contrast, we used a Friedman test to analyze the questionnaire data. We also report effect sizes of all statistically significant results: eta-squared ($\eta^2$) for ANOVA and Kendall's $W$ for Friedman test [3]. Eta-squared uses Cohen's interpretation [21], where 0.01 constitutes a small, 0.06 constitutes a medium, and over 0.14 constitutes a large effect. Kendall's $W$ uses a different interpretation by Cohen [21], where 0.1 constitutes a small, 0.3 constitutes a medium, and over 0.5 constitutes a large effect. There were no significant effects of the order of conditions on the dependent variables ($p > .8$), which suggests that counterbalancing worked [50, pp. 177–180].




(a) Task completion time (seconds)

(b) Actions per task

Fig. 8. Average task completion time and actions per task for all methods in the two settings (stationary and mobile). Error bars represent ±1 standard deviation (SD).

#### 4.7.1 *Task Completion Time.*

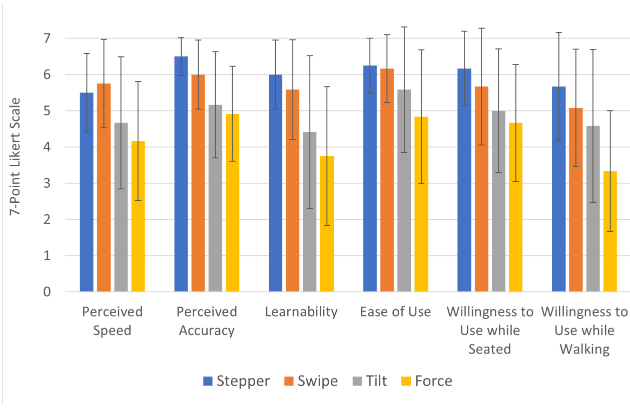
An ANOVA identified a significant effect of method on task completion time ($F_{3,33} = 23.76, p < .000005, \eta^2 = .07$)[5]. An ANOVA failed to identify a significant effect of setting ($F_{1,11} = 3.14, p = .10$). There was also no significant method × setting interaction effect ($F_{3,33} = 1.17, p = .33$). A Tukey-Kramer Multiple-Comparison test revealed that swipe was significantly faster than all other methods. Figure 8a illustrates average task completion time for the four methods in both settings.

#### 4.7.2 *Actions per Task.*

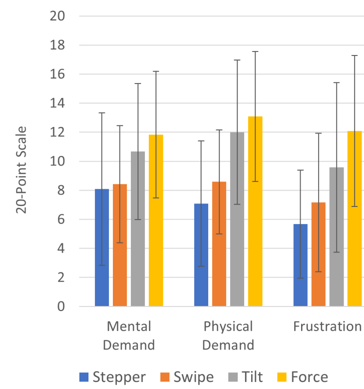
An ANOVA identified a significant effect of method on actions per task ($F_{3,33} = 14.71, p = .000003, \eta^2 = .18$). An ANOVA also identified a significant effect of setting ($F_{1,11} = 8.02, p = .016, \eta^2 = .003$). However, there was no significant method × setting interaction effect ($F_{3,33} = 1.28, p = .30$). A Tukey-Kramer Multiple-Comparison test revealed that stepper and

[5] The $p$ value was too small for the NCSS and SPSS statistical software to display the exact value as they and most other statistical software display values up to six decimal places.

tilt required significantly fewer actions per task compared to swipe and force. Figure 8b illustrates average actions per task for the four methods in both settings.




(a) Usability questionnaire




(b) Partial NASA-TLX questionnaire

Fig. 9. Average user ratings of the four methods on a 7-point Likert scale, where "1" to "7" represented "strongly disagree" to "strongly agree" (left) and on NASA-TLX's 20-point scale, where "1" to "20" represented "no demand" to "extreme demand" (right). Error bars represent ±1 standard deviation (SD).

### 4.8 User Feedback

We used a non-parametric Friedman test to analyze the questionnaire data. Here, we present raw TLX scores by analyzing the sub-scales individually, which is a common modification made to NASA-TLX [36]. Note that the evidence is inconclusive about whether raw TLX is more sensitive, less sensitive, or equally sensitive compared to the original version [36].

*4.8.1 Perceived Performance and Preference.* A Friedman test identified a significant effect of method on perceived speed ($\chi^2 = 9.31, df = 3, p = .025, W = .26$), perceived accuracy ($\chi^2 = 20.40, df = 3, p = .0001, W = .57$), learnability ($\chi^2 = 15.61, df = 3, p = .001, W = .43$), ease of use ($\chi^2 = 18.47, df = 3, p = .0003, W = .51$), and willingness to use in both stationary ($\chi^2 = 10.07, df = 3, p = .01, W = .28$) and mobile settings ($\chi^2 = 12.49, df = 3, p < .006, W = .35$). Participants rated picker and swipe substantially higher in all aspects compared to tilt and force. Force received the poorest ratings of all methods. Figure 9a illustrates average user ratings of the four methods.

*4.8.2 Mental and Physical Demand.* A Friedman test identified a significant effect of method on mental demand ($\chi^2 = 11.65, df = 3, p = .009, W = .32$), physical demand ($\chi^2 = 17.88, df = 3, p = .0004, W = .5$), and frustration ($\chi^2 = 19.41, df = 3, p = .0002, W = .54$). Participants found picker and swipe to be the least mentally and physically demanding compared to tilt and force. They also found the former methods to be the least frustrating. Force was rated substantially higher in terms of mental and physical demand, as well as frustration, compared to the other methods. Figure 9b illustrates average user ratings of the four methods.

## 5 DISCUSSION

Results revealed that swipe was the fastest of all methods in both stationary and mobile settings. The average task completion time for stepper, swipe, tilt, and force were 4.31 (SD = 1.8), 3.44 (SD = 3.7), 4.97 (SD = 2.6), and 5.42 (SD = 3.7) seconds, respectively. On average, swipe was about

30–45% faster than the other methods, regardless of the fact that it required significantly more actions per task than picker and tilt. This is likely because participants found performing swipes significantly easier than the other methods (Fig. 9). They also felt that performing swipes required less cognitive and physical demand than tilt and force, and caused less frustration (Fig. 9b). They found both stepper and swipe to be fast, accurate, and easy to use. There was no significant effect of method × setting on task completion time, which indicates towards the possibility that the performance of each method was similar across settings. This is interesting since prior studies reported performance decay when walking and interacting with a mobile device at the same time [4, 16]. This could be because of the slower pace of walking, and the use of a treadmill for walking since it did not require navigation. In real-world scenarios, users are forced to split their attention between the surroundings and the tasks on mobile devices to keep informed about the changing ambient environment [4]. Relevantly, a recent study with a similar experimental setup also failed to find a statistically significant difference in performance between input tasks in stationary and mobile settings [44]. On average, stepper was faster than tilt and force, however this difference was not statistically significant.

There was a significant effect of setting on actions per task, but no significant effect of method × setting. Which suggests that all methods suffered in terms of actions per task in mobile setting. Participants most likely performed more incorrect actions while walking, requiring corrective action, resulting in added actions per task. Interestingly, stepper took fewer actions than the other methods. Average actions per task for stepper, swipe, tilt, and force were 1.69 (SD = 1.2), 3.69 (SD = 3.5), 2.35 (SD = 1.5), and 4.81 (SD = 4.9), respectively. We speculate this is because, users spun the number wheel then waited until it slowed down, rather than a burst of repetitive spins. This also explains the higher task completion time for the method (Fig. 8).

The effects of method on the dependent variables yielded medium–large effect sizes, while the effect sizes of all statistically significant questionnaire data were large, indicating strong relationships between the examined variables. However, the effects of setting yielded a small effect size, hence we recommend caution in interpreting this result.

## 6 EXPERIMENT 2: INDIVIDUAL VS. INLINE

We conducted a second study to investigate whether the performance of the four methods differ when working with individual values and values embedded in text (i.e., inline values). The purpose was to find out whether the increased visual scan time and the physical and cognitive loads involved in editing inline numeric values affect the performance of the examined methods. Inline values usually require extra time to locate and select due to the surrounding text and the "fat-finger problem" [65], respectively. The study protocol was reviewed and approved by the Institutional Review Board (IRB). The study was completed before the World Health Organization declared the outbreak of COVID-19 a pandemic.

### 6.1 Apparatus

We used the same apparatus as the first study (Section 4.1).

### 6.2 Participants

Twelve participants voluntarily took part in the study. None of them participated in the pilots or the first study. Their age ranged from 20 to 32 years (M = 23.36, SD = 4.0). Two of them identified themselves as female and ten identified as male. Ten of them were right-handed, one was left-handed, and one was ambidextrous. All right-handed and ambidextrous participants chose to wear the device on their left hand and interact with the device using the right hand, while the left-handed participant chose to wear it on the right hand and use the other hand for interaction. They all were experienced smartphone users (M = 7.62 years of experience, SD = 1.9). Three of them were smartwatch owners (M = 1.8 years of ownership, SD = 0.8). Five of them also had experience using force as an input modality on Apple iPhone devices (M = 1.9 years of experience, SD = 1.7). They all received U.S. $10 for participating in the study.

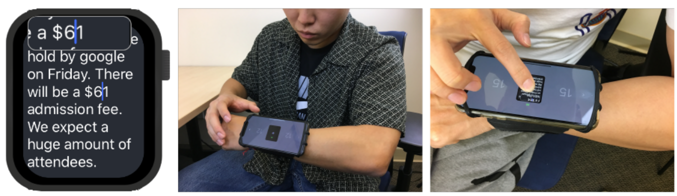



Fig. 10. A numeric value embedded in text (left) and two participants taking part in the study by changing an individual value and a value embedded in text, respectively (right).

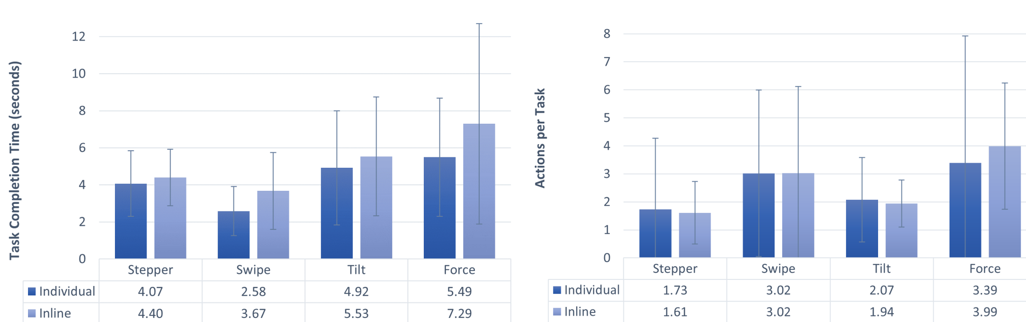
## 6.3 Design

We used a within-subjects design for the study, where the independent variables were *placement* and *method*, and the dependent variables were the same performance metrics as the first study (Section 4.3). We also used the same questionnaires. In summary, the design was as follows.

> 12 participants ×
> 2 placements (individual and inline, *counterbalanced*) ×
> 4 methods (the default stepper, swipe, tilt, and force, *counterbalanced*) ×
> 15 random two-digit values between 9 and 100
> = 1,440 numeric values in total, excluding practices.

## 6.4 Procedure

The study used the same procedure (Section 4.6) and experimental tasks (Section 4.4) as the first study, with the *setting* independent variable replaced with *placement*.




(a) Task completion time (seconds)

(b) Actions per task

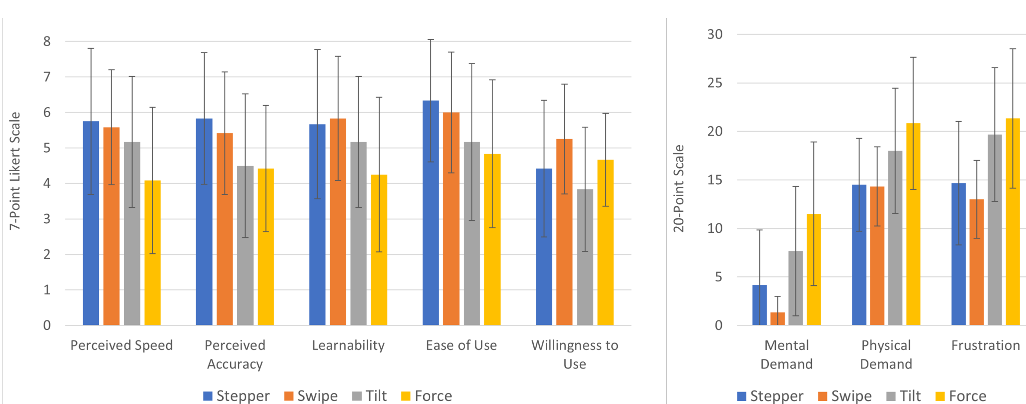
Fig. 11. Average task completion time and actions per task for all methods in the two settings (individual and inline). Error bars represent ±1 standard deviation (SD).

## 6.5 Results

A Shapiro-Wilk test revealed that the response variable residuals are normally distributed. A Mauchly's test indicated that the variances of populations are equal. Thus, we used a repeated-measures ANOVA to analyze the quantitative data. In contrast, we used a Friedman test to analyze the questionnaire data. We also report effect sizes of all statistically significant results: eta-squared ($\eta^2$) for ANOVA and Kendall's $W$ for Friedman test [3]. Eta-squared uses Cohen's interpretation [21] where 0.01 constitutes a small, 0.06 constitutes a medium, and > 0.14 constitutes a large effect. Kendall's $W$ uses a different interpretation by Cohen [21], where 0.1 constitutes a small, 0.3 constitutes a medium, and > 0.5 constitutes a large effect. There were no significant effects of the order of conditions on the dependent variables ($p > .75$), which suggests that counterbalancing worked [50, pp. 177–180].

*6.5.1 Task Completion Time.* An ANOVA identified a significant effect of method on task completion time ($F_{3,33} = 69.88, p < .000005, \eta^2 = .23$)[6]. An ANOVA also identified a significant effect of placement ($F_{1,11} = 22.37, p = .0006, \eta^2 = .03$). There was also a significant method × placement interaction effect ($F_{3,33} = 4.60, p = .0081, \eta^2 = .01$). A Tukey-Kramer Multiple-Comparison test revealed that all methods were significantly different from one another. Swipe was significantly faster and force was significantly slower than all other methods. Figure 11a illustrates average task completion time for the four methods in both placements.

*6.5.2 Actions per Task.* An ANOVA identified a significant effect of method on actions per task ($F_{3,33} = 5.17, p = .004, \eta^2 = .12$). An ANOVA failed to identify a significant effect of placement ($F_{1,11} = 0.34, p = .57$). There was no method × placement interaction effect ($F_{3,33} = 1.55, p = .22$). A Tukey-Kramer Multiple-Comparison test revealed that stepper and force were significantly different from one another. Stepper required significantly fewer actions per task compared to force. The other two methods were comparable. Figure 11b illustrates average actions per task for the four methods in both placements.




(a) Usability questionnaire (b) Partial NASA-TLX questionnaire

Fig. 12. Average user ratings of the four methods on a 7-point Likert scale, where "1" to "7" represented "strongly disagree" to "strongly agree" (left) and on NASA-TLX's 20-point scale, where "1" to "20" represented "no demand" to "extreme demand" (right). Error bars represent ±1 standard deviation (SD).

[6]The $p$ value was too small for the NCSS and SPSS statistical software to display the exact value as they and most other statistical software display values up to six decimal places.

### 6.6 User Feedback

We used a non-parametric Friedman test to analyze the questionnaire data. Like the previous study, we present raw TLX scores by analyzing the sub-scales individually [36].

*6.6.1 Perceived Performance and Preference.* A Friedman test identified a significant effect of method on ease of use ($\chi^2 = 11.78, df = 3, p = .008, W = .32$). However, no significant effect was identified on perceived speed ($\chi^2 = 7.8, df = 3, p = .05$), perceived accuracy ($\chi^2 = 7.73, df = 3, p = .05$), learnability ($\chi^2 = 7.19, df = 3, p = .06$), or willingness to use ($\chi^2 = 10.07, df = 3, p < .05$). Figure 12a illustrates average user ratings of the four methods.

*6.6.2 Mental and Physical Demand.* A Friedman test identified a significant effect of method on mental demand ($\chi^2 = 11.64, df = 3, p = .009, W = .32$), physical demand ($\chi^2 = 16.10, df = 3, p = .001, W = .45$), and frustration ($\chi^2 = 12.57, df = 3, p = .006, W = .35$). Participants found picker and swipe to be the least mentally and physically demanding compared to tilt and force. They also found the former methods to be the least frustrating. Force was rated substantially higher in terms of mental and physical demand, as well frustration compared to the other methods. Figure 12b illustrates average user ratings of the four methods.

## 7 DISCUSSION

The findings of this study are comparable to the first study. Swipe was significantly faster than the other methods with both individual and inline numbers. Average task completion time for stepper, swipe, tilt, and force were 4.23 (SD = 1.5), 3.13 (SD = 1.7), 5.23 (SD = 2.9), and 6.39 (SD = 4.3) seconds, respectively. Also, stepper and force required the least and the most number of actions, respectively. Average actions per task for stepper, swipe, tilt, and force were 1.67 (SD = 1.2), 3.02 (SD = 2.5), 2.01 (SD = 1.2), and 3.69 (SD = 4.5), respectively.

Qualitative results are also similar. Participants found both stepper and swipe significantly easier to use than the other methods. They found picker and swipe to be the least mentally and physically demanding compared to tilt and force. They also found swipe to be the least frustrating. These findings establishes the swipe-based method as a more effective input stepper for smartwatches. However, comparing Fig. 9b and Fig. 12b one can see that the methods were rated relatively poorly on NASA-TLX scale in the second study. This is expected since the inline placement of the numeric values required users to first locate and navigate to the value, then edit it, which required additional effort.

The effects of method on the dependent variables yielded medium–large effect sizes, while the effect sizes of all statistically significant questionnaire data were large, indicating strong relationships between the examined variables. However, the effects of placement yielded a small effect size, hence we recommend caution in interpreting this result.

### 7.1 Design Recommendations

Based on the results of the two studies and user feedback, we recommend using a hybrid of input stepper and the swipe-based method to enable selection of numbers with short edit distances with flicks (e.g., changing “12:15 pm” to “12:30 pm”) and long edit distances with swipes and swipe-and-hold gestures (e.g., changing “15%” to “70%”). We also recommend automatically slowing down the spinning rate when the number reaches a probable value for easier selection. A prior work [58] showed that it is often possible to predict the intended value through contextual awareness and even by using simple rules and patterns (for example, “12:30 pm” is more probable than “12:22 pm”). It may also be effective to enable users to select the method they prefer the most for picking numbers.

## 8 GENERALIZABILITY

Although it is relatively common to use smartphones or tablets to study interactions with smartwatches because of the technological limitations of current smartwatches (e.g., [18, 46, 57]), due to the absence of empirical evidence, it is unclear whether the performance recorded on a simulated smartwatch is generalizable to actual smartwatches. Hence, to increase the external validity of the work, we replicated not only the interface but also the holding position and posture of a smartwatch (Fig. 6). In all evaluations, participants wore the simulated smartwatch (i.e., the smartphone) on their wrist and interacted with it as one would with an actual smartwatch. Relevantly, a prior work reported that text entry performances with similar sized virtual keyboards on an actual smartwatch and a simulated smartwatch on a smartphone were not significantly different in terms of entry speed and accuracy [70].

## 9 CONCLUSION

We presented three new methods for number picking on smartwatches by performing directional swipes, twisting the wrist, and varying contact force on the screen. Unlike the default number picker, the proposed methods enable users to actively switch between slow-and-steady and fast-and-continuous increments and decrements. We evaluated these methods in two user studies, exploring stationary vs. mobile settings and individual vs. inline number editing, respectively. In both studies, the swipe-based method yielded a significantly faster input rate. Participants also found the method fast, accurate, and the least mentally and physically demanding. Accuracy rates were comparable among the conditions. These results establish swipe as an effective number picking method on smartwatches.

## 10 FUTURE WORK

In the future, we will evaluate the proposed number pickers on larger touchscreen-based devices, particularly smartphones and tablets. We will also design additional methods that exploit the crown and the bezel of a smartwatch for number picking.

## ACKNOWLEDGMENTS

This work was supported in part by a UC Merced Academic Senate Faculty Research Grant and a Hellman Fellows Award.

## REFERENCES

[1] Youngseok Ahn, Sungjae Hwang, HyunGook Yoon, Junghyeon Gim, and Jung-hee 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] Yomna Aly, Alan Yusheng Wu, Cosmin Munteanu, Molly Wei, and Stefania Raimondo. 2016. Spin-lock Gesture Authentication for Mobile Devices. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct, MobileHCI 2016*. 775–782. https://doi.org/10.1145/2957265.2961863

[3] Ahmed Sabbir Arif. 2021. Statistical Grounding. In *Intelligent Computing for Interactive System Design: Statistics, Digital Signal Processing, and Machine Learning in Practice* (1 ed.). Association for Computing Machinery, New York, NY, USA, 59–99. https://doi.org/10.1145/3447404.3447410

[4] Ahmed Sabbir Arif, Benedikt Iltisberger, and Wolfgang Stuerzlinger. 2011. Extending Mobile User Ambient Awareness for Nomadic Text Entry. In *Proceedings of the 23rd Australian Computer-Human Interaction Conference (OzCHI '11)*. ACM, New York, NY, USA, 21–30. https://doi.org/10.1145/2071536.2071539

[5] Ahmed Sabbir Arif, Roozbeh Manshaei, Sean Delong, Brien East, Matthew Kyan, and Ali Mazalek. 2016. Sparse Tangibles: Collaborative Exploration of Gene Networks Using Active Tangibles and Interactive Tabletops. In *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '16)*. ACM, New York, NY, USA, 287–295. https://doi.org/10.1145/2839462.2839500 event-place: Eindhoven, Netherlands.

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

[7] Ahmed Sabbir Arif, Ali Mazalek, and Wolfgang Stuerzlinger. 2014. The Use of Pseudo Pressure in Authenticating Smartphone Users. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MOBIQUITOUS '14)*. ICST, Brussels, Belgium, 151–160. https://doi.org/10.4108/icst.mobiquitous.2014.257919

[8] 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

[9] Leon Barnard, Ji Soo Yi, Julie A. Jacko, and Andrew Sears. 2005. An Empirical Comparison of Use-in-Motion Evaluation Scenarios for Mobile Computing Devices. *International Journal of Human-Computer Studies* 62, 4 (April 2005), 487–520. https://doi.org/10.1016/j.ijhcs.2004.12.002

[10] Alex Bellos. 2014. The World's Most Popular Numbers [Excerpt]. https://www.scientificamerican.com/article/most-popular-numbers-grapes-of-math/

[11] Joanna Bergstrom-Lehtovirta, Antti Oulasvirta, and Stephen Brewster. 2011. The Effects of Walking Speed on Target Acquisition on a Touchscreen Interface. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. Association for Computing Machinery, Stockholm, Sweden, 143–146. https://doi.org/10.1145/2037373.2037396

[12] John Brooke. 1996. SUS - a Quick and Dirty Usability Scale. *Usability Evaluation in Industry* 189, 194 (1996), 4–7. https://doi.org/10.1371/journal.pone.0170531

[13] Caroline Cakebread. 2017. Here's How People Are Using Their Smartwatches. https://www.businessinsider.com/most-used-smartwatch-features-chart-2017-8

[14] Géry Casiez, Daniel Vogel, Ravin Balakrishnan, and Andy Cockburn. 2008. The Impact of Control-Display Gain on User Performance in Pointing Tasks. *Human–Computer Interaction* 23, 3 (Aug. 2008), 215–250. https://doi.org/10.1080/07370020802278163 Publisher: Taylor & Francis _eprint: https://www.tandfonline.com/doi/pdf/10.1080/07370020802278163.

[15] Steven J. Castellucci, I. Scott MacKenzie, Mudit Misra, Laxmi Pandey, and Ahmed Sabbir Arif. 2019. TiltWriter: Design and Evaluation of a No-Touch Tilt-Based Text Entry Method for Handheld Devices. In *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia (MUM '19)*. ACM, 1–8. https://doi.org/10.1145/3365610.3365629 Article 7.

[16] A. Chamberlain and R. Kalawsky. 2004. A Comparative Investigation into Two Pointing Systems for Use with Wearable Computers While Mobile. In *Eighth International Symposium on Wearable Computers*, Vol. 1. 110–117. https://doi.org/10.1109/ISWC.2004.1 ISSN: 1530-0811.

[17] Youli Chang, Sehi L'Yi, Kyle Koh, and Jinwook Seo. 2015. *Understanding Users' Touch Behavior on Large Mobile Touch-Screens and Assisted Targeting by Tilting Gesture.* Association for Computing Machinery, New York, NY, USA, 1499–1508. https://doi.org/10.1145/2702123.2702425

[18] 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, 615–620. https://doi.org/10.1145/2642918.2647354

[19] Xiang 'Anthony' Chen, Tovi Grossman, Daniel J. Wigdor, and George Fitzmaurice. 2014. Duet: Exploring Joint Interactions on a Smart Phone and a Smart Watch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. Association for Computing Machinery, New York, NY, USA, 159–168. https://doi.org/10.1145/2556288.2556955

[20] A. Cockburn, D. Ahlström, and C. Gutwin. 2012. Understanding Performance in Touch Selections: Tap, Drag and Radial Pointing Drag with Finger, Stylus and Mouse. *International Journal of Human-Computer Studies* 70, 3 (March 2012), 218–233. https://doi.org/10.1016/j.ijhcs.2011.11.002

[21] J Cohen. 1988. *Statistical Power Analysis for the Behavioral Sciences* (2nd ed.). Lawrence Erlbaum, Hillsdale, NJ, USA. 283 pages. https://doi.org/10.1234/12345678

[22] Christian Corsten, Marcel Lahaye, Jan Borchers, and Simon Voelker. 2019. ForceRay: Extending Thumb Reach via Force Input Stabilizes Device Grip for Mobile Touch Input. *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19* Chi (2019), 1–12. https://doi.org/10.1145/3290605.3300442

[23] Christian Corsten, Simon Voelker, Andreas Link, and Jan Borchers. 2018. Use the Force Picker, Luke: Space-Efficient Value Input on Force-Sensitive Mobile Touchscreens. In *Conference on Human Factors in Computing Systems - Proceedings*, Vol. 2018-April. https://doi.org/10.1145/3173574.3174235

[24] 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.

[25] Sean DeLong, Ahmed Sabbir Arif, and Ali Mazalek. 2019. Design and Evaluation of Graphical Feedback on Tangible Interactions in a Low-Resolution Edge Display. ACM, 8. https://doi.org/10.1145/3321335.3324954

[26] Michael Dickens. 2011. Which Numbers are the Most Common? https://mathematicalmulticore.wordpress.com/2011/02/04/which-numbers-are-the-most-common/

[27] Mark D. Dunlop, Marc Roper, and Gennaro Imperatore. 2017. Text Entry Tap Accuracy and Exploration of Tilt Controlled Layered Interaction on Smartwatches. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '17)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3098279.3098560

[28] Lisa Eadicicco. 2015. New Details About The Screen For Apple's Next iPhone May Have Just Leaked. https://www.businessinsider.com/iphone-6s-rumors-screen-2015-1

[29] Brien East, Sean DeLong, Roozbeh Manshaei, Ahmed Arif, and Ali Mazalek. 2016. Actibles: Open Source Active Tangibles. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces - ISS '16*. ACM Press, Niagara Falls, Ontario, Canada, 469–472. https://doi.org/10.1145/2992154.2996874

[30] Angus Graeme Forbes and Tony Fast. [n.d.]. The Natural Materials Browser: Using a Tablet Interface for Exploring Volumetric Materials Science Datasets. ([n. d.]), 2.

[31] Jun Gong, Zheer Xu, Qifan Guo, Teddy Seyed, Xiang 'Anthony' Chen, Xiaojun Bi, and Xing-Dong Yang. 2018. WrisText: One-Handed Text Entry on Smartwatch Using Wrist Gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 181:1–181:14. https://doi.org/10.1145/3173574.3173755 event-place: Montreal QC, Canada.

[32] Mitchell Gordon, Tom Ouyang, and Shumin Zhai. 2016. WatchWriter: Tap and Gesture Typing on a Smartwatch Miniature Keyboard with Statistical Decoding. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 3817–3821. https://doi.org/10.1145/2858036.2858242 event-place: San Jose, California, USA.

[33] Jens Grubert, Matthias Heinisch, Aaron Quigley, and Dieter Schmalstieg. 2015. MultiFi: Multi Fidelity Interaction with Displays On and Around the Body. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3933–3942. https://doi.org/10.1145/2702123.2702331

[34] Aakar Gupta and Ravin Balakrishnan. 2016. DualKey: Miniature Screen Text Entry via Finger Identification. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. Association for Computing Machinery, San Jose, California, USA, 59–70. https://doi.org/10.1145/2858036.2858052

[35] 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.

[36] Sandra G. Hart. 2006. NASA-task Load Index (NASA-TLX); 20 Years Later. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 50. Sage publications Sage CA: Los Angeles, CA, 904–908.

[37] Uta Hinrichs and Sheelagh Carpendale. 2011. Gestures in The Wild: Studying Multi-Touch Gesture Sequences on Interactive Tabletop Exhibits. In *Conference on Human Factors in Computing Systems - Proceedings*. 3023–3032. https://doi.org/10.1145/1978942.1979391

[38] Jonggi Hong, Seongkook Heo, Poika Isokoski, and Geehyuk Lee. 2015. SplitBoard: A Simple Split Soft Keyboard for Wristwatch-Sized Touch Screens. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1233–1236. https://doi.org/10.1145/2702123.2702273 event-place: Seoul, Republic of Korea.

[39] T. Kaaresoja and J. Linjama. 2005. Perception of Short Tactile Pulses Generated by a Vibration Motor in a Mobile Phone. In *First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics Conference*. 471–472. https://doi.org/10.1109/WHC.2005.103

[40] Rushil Khurana, Nikola Banovic, and Kent Lyons. 2018. In Only 3 Minutes: Perceived Exertion Limits of Smartwatch Use. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers* (Singapore, Singapore) *(ISWC '18)*. Association for Computing Machinery, New York, NY, USA, 208–211. https://doi.org/10.1145/3267242.3267285

[41] Jungsoo Kim, Jiasheng He, Kent Lyons, and Thad Starner. 2007. The Gesture Watch: A Wireless Contact-free Gesture based Wrist Interface. In *2007 11th IEEE International Symposium on Wearable Computers*. IEEE, 15–22.

[42] Andreas Komninos and Mark Dunlop. 2014. Text Input on a Smart Watch. *IEEE Pervasive Computing* 13, 4 (Oct. 2014), 50–58. https://doi.org/10.1109/MPRV.2014.77 Conference Name: IEEE Pervasive Computing.

[43] Sven Kratz and Michael Rohs. 2009. HoverFlow: Exploring Around-Device Interaction with Ir Distance Sensors. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '09)*. Association for Computing Machinery, New York, NY, USA, 1–4. https://doi.org/10.1145/1613858.

1613912

[44] Satvik Kulshreshtha and Ahmed Sabbir Arif. 2020. Text Entry Performance on an Expandable Socket Attached Smartphone in Stationary and Mobile Settings. In *Advances in Usability and User Experience (Advances in Intelligent Systems and Computing)*, Tareq Ahram and Christianne Falcão (Eds.). Springer International Publishing, Cham, 207–217. https://doi.org/10.1007/978-3-030-19135-1_21

[45] 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*. 1–11.

[46] 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

[47] James R. Lewis. 2018. The System Usability Scale: Past, Present, and Future. *International Journal of Human–Computer Interaction* 34, 7 (July 2018), 577–590. https://doi.org/10.1080/10447318.2018.1455307

[48] O J Lewis, R J Hamshere, and T M Bucknill. 1970. The Anatomy of the Wrist Joint. *Journal of Anatomy* 106, Pt 3 (May 1970), 539–552. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1233428/

[49] Christian Loclair, Sean Gustafson, and Patrick Baudisch. 2010. PinchWatch: A Wearable Device for One-Handed Microinteractions. In *Proceedings of the 12th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '10)*. Association for Computing Machinery.

[50] I. Scott MacKenzie. 2013. *Human-Computer Interaction: An Empirical Research Perspective* (first edition ed.). Morgan Kaufmann is an imprint of Elsevier, Amsterdam.

[51] Meghna Mehta, Ahmed Sabbir Arif, Apurva Gupta, Sean DeLong, Roozbeh Manshaei, Graceline Williams, Manasvi Lalwani, Sanjay Chandrasekharan, and Ali Mazalek. 2016. Active Pathways: Using Active Tangibles and Interactive Tabletops for Collaborative Modeling in Systems Biology. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces (ISS '16)*. ACM, New York, NY, USA, 129–138. https://doi.org/10.1145/2992154.2992176 event-place: Niagara Falls, Ontario, Canada.

[52] Mudit Misra, Elena Márquez Segura, and Ahmed Sabbir Arif. 2019. Exploring the Pace of an Endless Runner Game in Stationary and Mobile Settings. In *Extended Abstracts of the Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts (CHI PLAY '19 Extended Abstracts)*. Association for Computing Machinery, Barcelona, Spain, 543–550. https://doi.org/10.1145/3341215.3356256

[53] Sachi Mizobuchi, Mark Chignell, and David Newton. 2005. Mobile Text Entry: Relationship Between Walking Speed and Text Input Task Difficulty. In *Proceedings of the 7th international conference on Human computer interaction with mobile devices & services (MobileHCI '05)*. Association for Computing Machinery, Salzburg, Austria, 122–128. https://doi.org/10.1145/1085777.1085798

[54] Fan Mo and Jia Zhou. 2020. Adapting Smartwatch Interfaces to Hand Gestures During Movements: Offset Models and the C-Shaped Pattern of Tapping. *Journal of Ambient Intelligence and Humanized Computing* (Sept. 2020). https://doi.org/10.1007/s12652-020-02545-3

[55] NASA. 1986. *NASA Task Load Index (TLX) V 1.0: Paper and Pencil Package*. Technical Report. Human Performance Research Group, NASA Ames Research Center, Moffett Field, CA, USA.

[56] Yusuke Niiro, Marcelo Kallmann, and Ahmed Sabbir Arif. 2019. An Experimental Comparison of Touch and Pen Gestures on a Vertical Display. In *Proceedings of the 8th ACM International Symposium on Pervasive Displays (PerDis '19)*. Association for Computing Machinery, New York, NY, USA, 1–6. https://doi.org/10.1145/3321335.3324936

[57] 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.

[58] Laxmi Pandey, Azar Alizadeh, and Ahmed Sabbir Arif. 2020. Enabling Predictive Number Entry and Editing on Touchscreen-Based Mobile Devices. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval (CHIIR '20)*. Association for Computing Machinery, New York, NY, USA, 13–22. https://doi.org/10.1145/3343413.3377957

[59] Benjamin Poppinga, Alireza Sahami Shirazi, Niels Henze, Wilko Heuten, and Susanne Boll. 2014. Understanding Shortcut Gestures on Mobile Touch Devices. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices & Services* (Toronto, ON, Canada) *(MobileHCI '14)*. Association for Computing Machinery, New York, NY, USA, 173–182. https://doi.org/10.1145/2628363.2628378

[60] Gulnar Rakhmetulla and Ahmed Sabbir Arif. 2020. SwipeRing: Gesture Typing on Smartwatches Using a Segmented Qwerty Around the Bezel. https://openreview.net/forum?id=eVyaNs3sm8a

[61] Daniel Spelmezan, Caroline Appert, Olivier Chapuis, and Emmanuel Pietriga. 2013. Side Pressure for Bidirectional Navigation on Small Devices. (2013). https://doi.org/10.1145/2493190.2493199

[62] Jessica J Tran, Shari Trewin, Calvin Swart, Bonnie E John, and John C Thomas. 2013. Exploring Pinch and Spread Gestures on Mobile Devices. In *MobileHCI 2013 - Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 151–160. https://doi.org/10.1145/2493190.2493221

[63] Theophanis Tsandilas, Caroline Appert, Anastasia Bezerianos, and David Bonnet. 2014. Coordination of Tilt and Touch in One- and Two-Handed Use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 2001–2004. https://doi.org/10.1145/2556288.2557088

[64] Colton J. Turner, Barbara S. Chaparro, and Jibo He. 2018. Texting While Walking: Is It Possible with a Smartwatch? *Journal of Usability Studies* 13, 2 (Feb. 2018), 94–118.

[65] 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.

[66] Emily A. Vogels. 2020. About One-in-Five Americans Use a Smart Watch or Fitness Tracker. https://www.pewresearch.org/fact-tank/2020/01/09/about-one-in-five-americans-use-a-smart-watch-or-fitness-tracker/

[67] Katrin Wolf, Robert Schleicher, and Michael Rohs. 2014. Ergonomic Characteristics of Gestures for Front- and Back-of-Tablets Interaction with Grasping Hands. In *MobileHCI 2014 - Proceedings of the 16th ACM International Conference on Human-Computer Interaction with Mobile Devices and Services*. 453–458. https://doi.org/10.1145/2628363.2634214

[68] Chao Xu, Parth H. Pathak, and Prasant Mohapatra. 2015. Finger-Writing with Smartwatch: A Case for Finger and Hand Gesture Recognition Using Smartwatch. In *Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications* (Santa Fe, New Mexico, USA) *(HotMobile '15)*. Association for Computing Machinery, New York, NY, USA, 9–14. https://doi.org/10.1145/2699343.2699350

[69] 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.

[70] 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


Received July 2021; revised September 2021; accepted September 2021