

Investigating Tilt-based Gesture Keyboard Entry for Single-Handed Text Entry on Large Devices

Hui-Shyong Yeo¹, Xiao-Shen Phang², Steven J. Castellucci³, Per Ola Kristensson⁴
and Aaron Quigley¹

¹University of St Andrews, United Kingdom ²NEC Corporation of Malaysia, Malaysia

³York University, Canada ⁴University of Cambridge, United Kingdom

¹{hsy, aquigley}@st-andrews.ac.uk, ms.phang2@gmail.com, stevenc@yorku.ca, pok21@cam.ac.uk

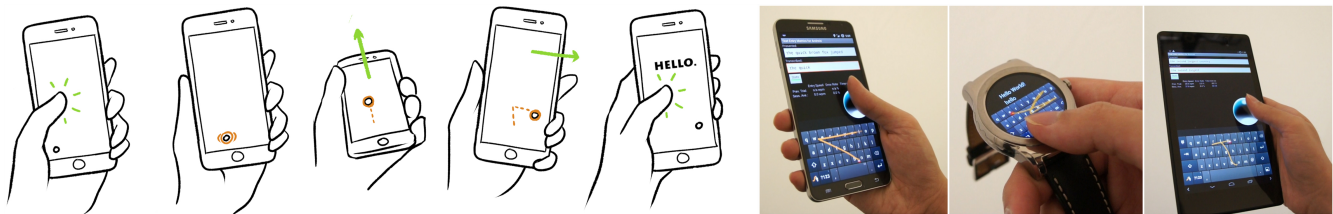


Figure 1. (left) SWiM technique: Tap and tilt the device to trace the word. (right) SWiM prototype on a phone, smartwatch and tablet.

ABSTRACT

The popularity of mobile devices with large screens is making single-handed interaction difficult. We propose and evaluate a novel design point around a tilt-based text entry technique which supports single handed usage. Our technique is based on the gesture keyboard (shape writing). However, instead of drawing gestures with a finger or stylus, users articulate a gesture by tilting the device. This can be especially useful when the user's other hand is otherwise encumbered or unavailable. We show that novice users achieve an entry rate of 15 words-per-minute (wpm) after minimal practice. A pilot longitudinal study reveals that a single participant achieved an entry rate of 32 wpm after approximate 90 minutes of practice. Our data indicate that tilt-based gesture keyboard entry enables walk-up use and provides a suitable text entry rate for occasional use and can act as a promising alternative to single-handed typing in certain situations.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): User interfaces—*Input devices and strategies*;

Author Keywords

Text entry; single-handed; tilt; shape writing; gesture keyboard; phablets.

INTRODUCTION

Screen sizes on mobile devices have been increasing, as shown by the recent popularity of phablets (e.g., 5.7" Galaxy Note 3 and 5.5" iPhone 6 Plus). However, it's difficult to hold a phablet firmly in one hand, let alone interact with it. This is due, in part, to the limited functional area of the thumb [4] which makes it difficult to reach all areas of the screen. In addition, unintended touches from the palm area may occur. These problems will be exacerbated if the trend of larger mobile phone screens continue. Yet, there are many occasions when the user's other hand is encumbered [22] or not available, such as due to a disability or when holding a bag, a cup of coffee, or an umbrella. Indeed such interaction is now commonplace, as an informal observation of people in public places showed 49% of people hold the phone using only one hand [15].

Mobile devices are often used for text-based communication. Analysts estimate the annual worldwide number of Short Message Service (SMS) text messages sent to be 6.5 trillion and the number of Over-the-top content (OTT) instant messages sent over the Internet (e.g., using WhatsApp) to be 10.3 trillion [27]. A study [1] showed that, even when walking or commuting, 85-88% of users continued typing with one hand while the other is not available. However, holding a phablet and typing with one hand can be difficult, if not impossible.

Azenkot and Zhai [3] explored different hand postures for typing on a phone and found that one-thumb typing can be quite fast, achieving 33.78 wpm, assuming no errors occur. They used a relatively small 4.3" phone screen, where there is no thumb reachability problem. In contrast, typing one-handed on a larger device requires changing or shuffling one's grip frequently, which is difficult and might result in accidentally dropping the device. It could also cause unintended touches,

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

CHI 2017, May 6-11, 2017, Denver, CO, USA.

Copyright © 2017 ACM ISBN 978-1-4503-4655-9/17/05 ...\$15.00.

<http://dx.doi.org/10.1145/3025453.3025520>

especially from the inner palm area of the hand near the thumb. To address this issue, developers and vendors often shrink the touchscreen keyboards to effectively extend the thumb's reach. This solution is far from perfect since shrinking the keyboard to one side requires the user to touch with the side of the thumb rather than the tip, hence, it can exacerbate occlusion and the fat finger [13, 14, 29] problems.

In this paper, we introduce *Shape Writing in Motion* (SWiM), a tilt-based gesture keyboard text entry technique that supports entering one word per gesture. As the name implies, it is based on the shape writing technique [17, 19, 40], commonly referred to as a gesture keyboard [18, 41], which we extend to support tilt. Instead of using a finger to draw over the on-screen keyboard, we use wrist motion of the dominant hand to perform shape writing. As previously noted, the dominant hand typically exhibits finer temporal and spatial resolutions [11]. Our tilt-based technique allows one-handed text entry and also lends itself as a suitable text entry method for remote controllers, wearable devices, and for virtual reality text entry, where users cannot see the keyboard input device.

We make the following contributions: 1) We propose and implement a technique (SWiM) for tilt-based single-handed text entry that supports mobile devices of various form factors including phone, tablet, and smartwatch. 2) We evaluate text entry performance of our technique using both a *transcription* and a *composition* task. The results show that SWiM is fast and easy to learn compared to many existing single-handed text entry techniques. In addition, we explore performance improvement over time via a longitudinal pilot study.

RELATED WORK

Many existing single-handed text entry methods [7, 9, 16, 28, 35, 37], while promising, suffer from inefficiencies. They only support entry of a single character for each defined action. This can be slow and tedious, especially when writing long messages. In contrast, SWiM produces a full word per gesture.

Tilt-based Mobile Interaction and Text Entry

Tilt has been frequently explored for mobile interaction. Rekimoto [25] and Weberg et al. [36] explored using a tilt signal to navigate menus and 2D maps. Several other studies [8, 12, 33] also explored single-handed interaction on mobile devices by combining motion and touch to 1) assist targeting and extend thumb's reach [8], 2) enhance navigation with quick command gestures [33], and 3) enable new types of motion gestures, such as tilt to zoom, pivot-to-lock, and shake to delete [12]. Although there have been many attempts to use tilt and tilt gestures for text entry, we have applied tilt to shape writing to address the single-handed text entry problem.

TiltType [23] was the first to explore text entry with a tilt technique. TiltType combines multiple buttons with the direction and the angle of tilt to disambiguate the intended letter, but no evaluation was performed. Later, TiltText [37] proposed a technique that combines tilt with the standard 12-key phone keypad to reduce the time required to disambiguate the letters on a single key. TiltText reported a novice speed and a last block speed of 7.42 wpm and 13.57 wpm, respectively. GestText [16] uses only the accelerometer found in a Wiimote and

converts the tilting action into a letter based on two proposed layouts. It reported 5.3 wpm on the fourth day for the fastest layout. Fitton et al. [9] used a tablet's built-in accelerometer to control a pointer over a virtual keyboard, with dwelling as the delimiter to achieve 10 wpm entry rate.

In the Unigesture approach, Sazawal et al. [28] proposed using a sequence of tilt gestures to infer a word. Input speed of roughly 4 wpm was achieved. This demonstrates that word-based input is not always faster than character-based input. Williamson et al. [38] presented a gestural text entry technique via continuous movements but no formal evaluation was conducted, except the author claimed that he was able to achieve 10–12 wpm. Castellucci et al. [7] incorporated Huffman codes with gestures for text entry, and reported an entry rate of 5.3 wpm. Rotext [35] proposed a one-dimensional rotational text entry technique that supports sight-free use by mapping the device orientation to an optimized layout. It achieves 14.2 wpm on the first session (sighted) and 12.6 wpm on the final session (sight-free). These techniques could, with practice, be used eyes-free. This would benefit mobile users, as their visual attention could return to navigating their environment and avoiding obstacles. SWiM is not capable of such input. However, in exchange for attending to the keyboard, the SWiM users would be able to use any word correction or completion features of the gesture keyboard.

Alternative Shape Writing Methods

Shape writing (gesture keyboard) [17, 19] enables fast text entry by allowing users to draw the shape of a word instead of typing each character. It gained popularity on touch screen devices but not in other input modalities. Recently, Vulture [21] explores shape writing in mid-air, and achieves 11.8 wpm (first session) and 20.6 wpm (tenth session). However, it requires a complex setup involving optical markers and is not practical for mobile computing. Kristensson [17] also suggested an alternative solution for shape writing that uses a phantom keyboard to reduce the fat finger and occlusion problem. Reyat et al. [26] evaluated real-world use of shape writing on touchscreen mobile phones. Users eventually reached an entry rate of nearly 40 wpm when using shape writer daily on their own phones. Gordon et al. [10] evaluated shape writing for touch-based smartwatch text entry and reported an entry rate of 24 wpm. Lastly, a similar technique to our approach is the SwiftKey Tilt [30]. However, it was an April Fools' joke and the prototype does not trace the word using tilt.

SYSTEM DESIGN

SWiM is a tilt-based text entry technique. Therefore our design is based on related tilt-based research on wrist and text entry. SWiM leverages the presence of accelerometers and gyroscopes in modern mobile devices. In addition to phones and tablets, SWiM could be included in many other devices, such as game controllers, TV remotes, head-mounted displays (HMD), and wearable devices.

Designing SWiM Interaction

Traditional shape writing involves the following procedure for the user to enter a word [19]: 1) Touch the key corresponding to the first letter in the word; 2) Draw a line over all successive

letters, in order; and 3) Release the touch. SWiM has a similar procedure but it is modified with tilt input: 1) Tilt the device so that the pointer (i.e., ball) is on the first letter, and tap the “chat head” (detailed below) to begin input; 2) Tilt the device so that the ball traces a path over all successive letters, in order; and 3) Tap the chat head again to end input. The overall steps are depicted in Figure 1 (left).

Leveraging the knowledge from previous tilt-based research, we employ position-control for the pointer in SWiM, as it is superior to velocity-control in many aspects [32]. Furthermore, we use only the pitch and roll, as the dexterity of yaw using one’s hand is very limited [24]. We limit the range of pitch and roll motion to ± 30 degrees from a rest position to i) improve screen visibility and ii) because wrist motion is most sensitive and precise within this range [24]. For example, when tilted 30 degrees left, the pointer will be on the left edge of the screen. Any further tilt will not move the pointer outside the screen. Finally, we use a linear mapping instead of a quadratic mapping because Rotext [35] showed that linear mapping is more accurate for text entry purposes, even though Rahman et al. [24] suggest that a quadratic mapping is better for general tilt-based selection and pointing.

Importantly, SWiM augments existing commercial touch-screen gesture keyboards, such as ShapeWriter, Google’s gesture keyboard, Swype [31], and SwiftKey-Flow [30]. SWiM injects touch events based on the position of the pointer. The system detects the start and end of a gesture when the user taps the chat head (a floating touch region similar to Facebook Messenger’s chat head or iPhone’s AssistiveTouch), as shown as the blue floating bubble in Figure 1 (right). This touch region can be easily relocated to a different location on the screen using a long drag. Thus, it supports either handedness and different hand grip postures. It can also be made semi-transparent with a minimal outline so that it does not occlude screen content.

An advantage of our tilt-based implementation is that it enables simple swipe gestures (e.g., swipe left to delete) in addition to the tilt-based shape writing. These simple swipe gestures have been shown to be beneficial [2], but supporting them was previously impossible, due to typical gesture keyboard would interpret the swipe as word input.

IMPLEMENTATION

During text entry, we retrieve the orientation of the device (pitch and roll) and map it to absolute tilt magnitude (degrees) and screen coordinates (x, y) for controlling a pointer. We then overlay the pointer (e.g., ball) and a touch region (chat head) on top of any application. The position of the pointer is smoothed using the one-Euro filter [5] to reduce jitter.

When a tap is detected on the chat head, we start injecting a touch-down event and touch-move event at the current pointer position. Therefore, a dragging action is seen when the device is being tilted, as shown in Figure 1. We then inject a touch-release event on the second tap. We intentionally did not constrain the ball inside the keyboard area (bottom half of the screen) because it would disable one of the useful features of the gesture keyboard: dragging out from the keyboard as a

shortcut to capitalize the letter. By default, the ball position is always initialized in the middle of the keyboard area when the device is held with a typical hand posture, inclined about 45 degrees. This rest position is user-customizable via a slider.

The remainder of the text entry process is handled by the shape writing keyboard itself. In general, the soft keyboard will use the dragged path to produce the most appropriate word [19], along with other highly probable words on the suggestion bar. Users can continue input, which implicitly selects the first suggested word and inserts a space. Alternatively, the user can select a word from the suggestion bar by using the ball as a pointer and performing a short click. If the desired word is not in the suggestion bar, users can swipe left on the chat head to delete the current word, and start over again. A swipe right gesture on the chat head will simulate pressing the ENTER key. While swipe up and swipe down are currently not assigned, they could be used for toggling the SHIFT key. In addition, the chat head can be easily extended to support more complex gestures such as swipe left-down, circle, or pigtail gestures.

EVALUATION

Our objective was to study the immediate usability of SWiM with minimal training, and how it compares to one-thumb typing and one-thumb shape writing on a large mobile device (e.g., a phablet). We conducted one study with a traditional transcription task and one study with a composition task [34], specifically the reply task. One participant from the first study also volunteered for a longitudinal study which consisted of five more sessions of the same transcription task.

Apparatus and Participants

The primary study was conducted on a 5.7" Samsung Galaxy Note 3 Android phone, as the device area represents a common phablet size in the market. For one-thumb typing and one-thumb shape writing, the keyboard was reduced in size and aligned right/left to ease the thumb’s reach, to simulate more realistic real world usage, and to reduce bias towards our technique. We used the Swype [31] keyboard throughout our study for consistency across all techniques. The keyboard dimensions were 7×4.2 cm (default) and 5.8×4.2 cm (reduced). We used Text Entry Metrics on Android (TEMA) [6] to present the phrases and log the transcribed text, along with measuring text entry metrics.

For the transcription task, we recruited 12 participants (3 females and one left-handed) from our university campus. Their ages ranged from 19 to 30 years (mean = 23.7, sd = 4). Two participants had experience with shape writing but none had used tilt for text entry before. Five of the participants are native English speakers and the rest can communicate in English fluently. Their self-rated English skills were between 4 and 7 (mean = 5.96, sd = 1.03) on a seven-point scale, where “1 - No English skills” to “7 - Native”.

For the composition task, we recruited 6 participants (all male, all right-handed) from our university campus. Ages ranged from 18 to 37 years (mean = 25.0, sd = 6.84). Three participants had experience with shape writing but none had used tilt for text entry before. Two of the participants are native English speakers and the rest can communicate in English

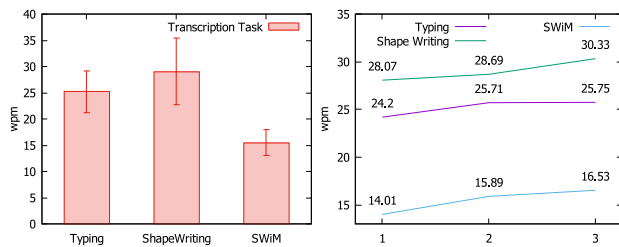


Figure 2. Transcription task: (left) Mean entry rate in wpm of 3 blocks in the first session. Errors bars equal ± 1 sd. (right) Mean entry rate in wpm of each of the three blocks in the first session.

fluently. Their self-rated English skills were between 6 and 7 (mean = 6.33, sd = 0.51) on the same scale as above.

Procedure

Before the experiment, we explained the tasks and demonstrated the three techniques: one-thumb typing, one-thumb shape writing, and SWiM. Participants were allowed 10 practice trials for each technique before proceeding with the experiment. Technique order was counter-balanced.

For the transcription task, participants were instructed to type the provided phrases as quickly and as accurately as possible. This is similar to previous text entry studies [39]. For the composition task, we used the recommended instruction in [34], which for completeness is quoted in full below: “Imagine you are using a mobile device and need to write a message. We want you to invent and type in a fictitious (but plausible) message. Use your imagination. If you are struggling for ideas, think about things you often write about using your own mobile device.” [34]

On average, the study took 60 minutes. Users were seated throughout the test and held the device with their dominant hand. A NASA-TLX questionnaire was administered to each user at the end of the study. For both studies, participants were compensated with a £10 Amazon voucher.

Design

For both studies, we used a within-subjects design with one independent variable, *technique*, with three levels: one-thumb typing, one-thumb shape writing and SWiM. Participants completed one session, consisting of the 3 techniques. For each technique, participants completed 3 blocks of 10 phrases. For the transcription task, participants entered 10 phrases, sampled randomly from the MacKenzie and Soukoreff phrase set [20]. This resulted in a total of 12 participants $\times$ 3 techniques $\times$

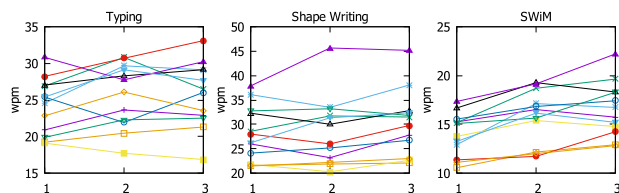


Figure 3. Entry rate of each technique in wpm for each of the twelve participants over three blocks of ten phrases in the first session.

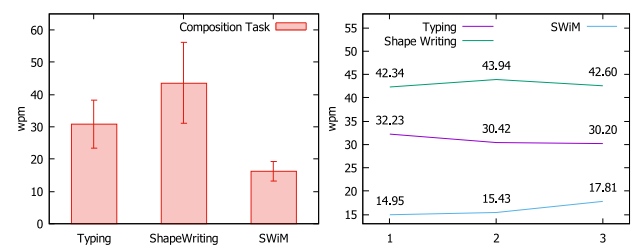


Figure 4. Composition task: (left) Mean entry rate in wpm of 3 blocks in the first session. Errors bars equal ± 1 sd. (right) Mean entry rate in wpm of each of the three blocks in the first session.

3 blocks $\times$ 10 phrases = 1080 transcribed phrases. For the composition task, participants entered 10 replies to messages from the Vertanen and Kristensson [34] corpus. This totaled 6 participants $\times$ 3 techniques $\times$ 3 blocks $\times$ 10 phrases = 540 transcribed phrases. For the longitudinal study, one volunteer continued with the transcription task for five more sessions. In total, this participant entered 180 phrases for each technique.

The dependent variables were entry rate and error rate. We measured entry rate in words per minute (wpm), where a word is defined as five consecutive characters including space. We measured error rate as the character error rate (CER), which is calculated as the minimum edit distance between the presented phrase and the transcribed text, divided by the number of characters in the presented phrase, and reported as a percent.

Auto correction was partially enabled for the typing technique. The keyboard would not automatically replace a wrong word with a corrected one after the space key is hit. However, the auto-corrected word will appear as the first word in the suggestion bar and participants were encouraged to use that to select the correct word. Nonetheless, users were also encouraged to type as quickly and as accurately as possible, rather than being careless and only relying on auto-correction.

Quantitative Results

Transcription Task

In the transcription task, participants achieved mean entry rates of 25.22 wpm (sd = 4.01), 29.03 wpm (sd = 6.34), and 15.48 wpm (sd = 2.46) in the typing, shape writing, and SWiM conditions, respectively. These results are summarized in Figure 2 (overall) and Figure 3 (per-user). Conducting a repeated-measures analysis of variance revealed that the effect of input technique on entry speed was significant ($F_{2,12} = 39.38$, $p < 0.0001$), with an observed power of 1.0. Tukey, Scheffé, LSD, and Bonferroni post-hoc tests all indicated significance only between the tilt technique and the other two techniques. Further our counterbalancing was effective as the order of input techniques was not significant ($F_{5,6} = 0.51$, $p > 0.05$).

Participants entered text accurately with all three techniques. The mean character error rates (CER) in the typing, shape writing, and SWiM conditions was 0.42% (sd = 0.40%), 0.67% (sd = 0.39%), and 0.88% (sd = 0.82%), respectively. Unlike entry rate, input technique did not have a significant effect on CER ($F_{2,12} = 2.88$, $p > 0.05$). Though, counterbalancing was again effective ($F_{5,6} = 0.51$, $p > 0.05$).

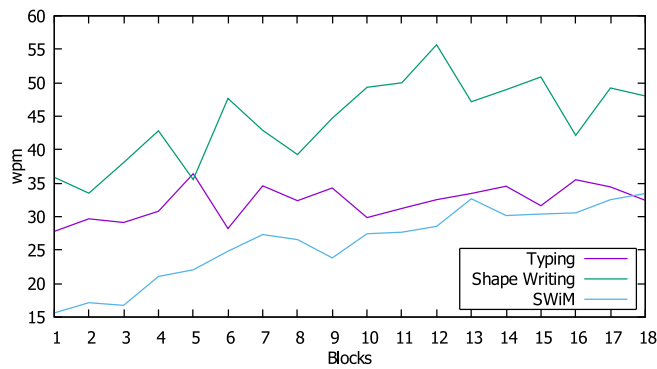


Figure 5. Pilot longitudinal study shows learning effect of P3 for extra five sessions, visualized over 18 blocks where each block has 10 phrases.

Composition Task (Reply)

In the composition task, participants achieved mean entry rates of 30.83 wpm (sd = 7.45), 43.59 wpm (sd = 12.54), and 16.28 wpm (sd = 2.93) in the typing, shape writing, and SWiM conditions, respectively. These results are summarized in Figure 4. Input technique again had a significant effect on entry rate ($F_{2,6} = 41.85$, $p < 0.0001$). Post-hoc analysis revealed significance between shape writing and SWiM. An LSD test did show significance between all three conditions, but LSD is a less powerful test. As with the previous study, counterbalancing was effective ($F_{2,3} = 0.39$, $p > 0.05$).

We analyzed the error rate in the composition task manually following the protocol introduced by Vertanen and Kristensson [34]. Mean CER in the typing, shape writing, and SWiM conditions was 0.78% (sd = 0.71%), 0.92% (sd = 0.76%), and 1.4% (sd = 0.66%), respectively. Input technique did not have a significant effect on CER ($F_{2,6} = 0.700$, $p > 0.05$). As with the previous study, counterbalancing was effective ($F_{2,3} = 0.39$, $p > 0.05$). Participants generally entered short replies with an average character count of 23.47 (Typing), 22.81 (Shape Writing), and 20.58 (SWiM).

Longitudinal Pilot

One participant (P3) from the transcription task volunteered to perform a 5-session longitudinal pilot study (Figure 5). In the first session, this participant achieved an entry rate of 27.79 wpm (typing), 35.89 wpm (shape writing) and 15.60 wpm (SWiM). By the sixth session, the same participant achieved 34.13 wpm (typing), 46.47 wpm (shape writing), and 32.18 wpm (SWiM). This is an improvement of 22.8%, 29.5%, and 106.3% for typing, shape writing and SWiM, respectively. We estimated approximately 90 minutes of usage for each technique. The first session took roughly 60 minutes for 3 techniques including briefing and resting time, while the following sessions were much faster.

Qualitative Results

Transcription Task

The NASA-TLX scores for all three conditions are shown in Figure 6 (lower scores are better). As expected, participants rated the novel technique as having a higher workload than the more familiar text entry methods. In the effort subscale,

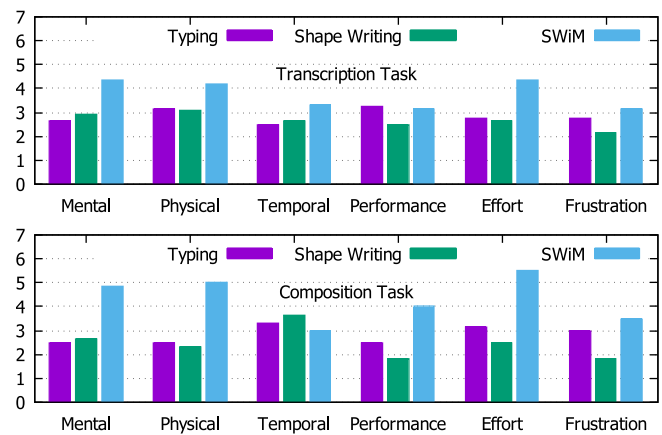


Figure 6. (Top) Transcription task: Average NASA-TLX scores from 12 participants. (Bottom) Composition task: Average NASA-TLX scores from 6 participants. Lower is better.

SWiM scored 4.3, shape writing scored 2.7, and typing scored 2.8. Interestingly, the scores for performance were similar. Shape writing scored best (2.5), followed by SWiM (3.2), and typing (3.3). It is very encouraging to have novice users rate SWiM as having similar self-rated performance as established techniques.

Due to the non-parametric nature of the scores, Friedman's test was used to compare ratings of the three within-subjects conditions. The test corrects for ties, of which there were many. Subsequent post-hoc comparisons using the Conover's test were conducted, when we found statistical significance.

The statistical analysis revealed significant differences amongst the techniques with respect to effort ($\chi^2 = 9.897$, $p < 0.01$, $df = 2$), mental ($\chi^2 = 10.844$, $p < 0.005$, $df = 2$), and physical ($\chi^2 = 6.0$, $p < 0.05$, $df = 2$). For both effort and mental subscales, the difference was between SWiM and the other two conditions. For the physical subscale, there was only a significant difference between SWiM and shape writing.

Composition Task

For the composition study, the NASA-TLX score is shown in Figure 6. As with the transcription task, the scores for SWiM were worse than the other two techniques. With the composition task though, the disparity in scores was larger. This is especially so for effort, where SWiM scored 5.5, versus 3.2 for shape writing, and 2.5 for typing. There were significant differences amongst the techniques with respect to effort ($\chi^2 = 9.364$, $p < 0.01$, $df = 2$), mental ($\chi^2 = 9.10$, $p < 0.05$, $df = 2$), performance ($\chi^2 = 6.952$, $p < 0.05$, $df = 2$), and physical ($\chi^2 = 7.60$, $p < 0.05$, $df = 2$) subscales. For both effort, mental, and physical subscales, the difference was between SWiM and the other two conditions. For the performance subscale, there was only a significant difference between SWiM and shape writing.

DISCUSSION

In the transcription task, participants' SWiM entry rates were about half of those for one-thumb typing and shape writing. The mean CER remained low (below 0.9%) for all techniques.

This suggests that they are all suitable for text entry tasks, ranging from casual messaging to more formal communication. Since all of our users have little to no experience with the tilt technique, we expected a steep learning curve. Indeed the NASA-TLX scores indicate SWiM is the most demanding of the three techniques. Yet, our results show that users were able to enter text using SWiM relatively well after minimal training. Figure 2 (transcription) and Figure 4 (composition) both show that SWiM's performance steadily grows over time, even with only 3 blocks of trials. SWiM's increasing entry rate is more obvious in the longitudinal pilot study results (Figure 5). In the composition task, entry rates for all techniques were higher than the transcription task.

It is not possible to directly compare performance figures across different studies as many variables change (participants, apparatus, task, etc.). However, as an *indication* of SWiM's relative performance compared to previous tilt-based text entry methods, we note that SWiM's mean entry rates surpass those obtainable from previously published tilt-based text entry methods. We believe that expert performance of SWiM can increase further, since SWiM is built on the principles of the gesture keyboard [19], which gradually transitions people from slow visually-guided closed-loop interaction to fast open-loop direct recall from motor memory. From our pilot longitudinal study, we can see that SWiM has a much higher improvement rate (106.3%) over the other two techniques (22.8% and 29.5%), which indicates that the other techniques' performance potential saturates faster.

Our observed one-thumb typing speed is slightly slower than the one reported by Azenkot et al. [3], probably since they assumed error-free input. We also again note that it is not possible to directly compare raw performance data points across different studies. For instance, our study used a larger phone, which means typing is impaired as a result of the thumb reach problem. Our entry rates for touch-based shape writing align well with Reyat et al.'s empirical study of shape writing "in the wild" [26].

We observed that participants tended to use a combination of ulnar/radial deviation and flexion/extension of the wrist when tilting the mobile device upwards or downwards. This caused the pointer to deviate in a diagonal direction. We could solve this problem by using a different mapping function.

We also observed that none of the users were able to grasp the phone firmly when typing with one-thumb, except when using SWiM. While there was no occurrence of the device dropping in our experiment, we can easily imagine it happening during actual use. Thus, SWiM has a clear advantage in terms of device grip robustness. During the one-thumb typing and shape writing study, there were a few accidental touches from participants' inner palms that triggered the BACK or the ENTER keys, skipping that trial. These erroneous data points were later removed before the statistical analyses.

Interestingly, P5's one thumb typing speed decreased from block 1 to block 3 (Figure 3) in the transcription task. As P5's palm size was very small, P5 struggled to reach for the keys on the other edge, and this might have lead to palm fatigue.

Another explanation is that typing was the last condition for P5. P5 said that she previously owned a large phone, but switched back to a smaller 5-inch iPhone because of this thumb's reach problem. Most participants mentioned that they primarily type with two hands, because one-handed typing on a large phone is difficult. These comments validate our motivation for this work. Also, several participants mentioned that they did not buy, and do not plan to buy, a phone larger than five inches because of this single-handed problem. After trying our new technique, they expressed an interest in using SWiM on their own phones, once an official version is available. This reinforces our view that SWiM opens up a new design point in the mobile text entry design space.

Finally, our tilt-based keyboard only requires a small area for touch detection. Since the pointer movement is directly proportional to the tilt magnitude, we can reduce the keyboard size so that active applications can use more screen space. This also means that SWiM is likely to be a suitable mobile text entry method for some wearable designs, such as digital pendants coupled with an optical see-through head-mounted display (HMD). Alternatively, we could make the keyboard semi-transparent. In addition, we believe our technique to be well suited for input in virtual reality (VR) applications. When wearing a VR headset, the user often cannot see the keyboard. Thus, a simple controller that supports tilt (e.g., Google VR controller, Oculus Touch) can use SWiM for text input.

Although not tested with users, one of the authors confirmed that SWiM works well on larger tablets (Nexus 7 and Nexus 10) and on a smartwatch (LG Urbane), as shown in the video figure accompanying this paper. One limitation is that the current version injects touch events into the keyboard, which requires root privileges. The advantage is that any existing commercial gesture keyboard can be used as a recognizer. The disadvantage is that existing commercial gesture keyboards are not optimized to handle a tilt signal.

CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a tilt-based text entry method that is easy to use with only one hand, and is usable even on large mobile devices. Our studies show that the technique has high immediate efficacy and is reasonably fast. As we emphasized, SWiM is not meant to replace single-handed or two-handed typing in all situations, but rather to facilitate text entry when only one hand is available and when using a large device (e.g., phablet) or small devices (e.g., smartwatch).

In the future, we plan to integrate our technique with a gesture keyboard and release it as an installable touchscreen keyboard application. We will continue to investigate which factors affect tilt-based shape writing and hope to improve SWiM over time. Lastly, we aim to use this technique on other devices, such as smartwatches, digital pendants, and head-mounted displays (HMD), which already have built-in tilt sensors.

ACKNOWLEDGEMENTS

The authors thank the reviewers for their constructive feedback. The research described here was supported by the University of St Andrews and the Scottish Informatics and Computer Science Alliance (SICSA).

REFERENCES

1. A.S. Arif. 2012. A survey on mobile text entry handedness: How do users input text on handheld devices while nomadic?. In *Intelligent Human Computer Interaction (IHCI), 2012 4th International Conference on*. 1–6. DOI: <http://dx.doi.org/10.1109/IHCI.2012.6481818>
2. Ahmed Sabbir Arif, Michel Pahud, Ken Hinckley, and Bill Buxton. 2014. Experimental Study of Stroke Shortcuts for a Touchscreen Keyboard with Gesture-redundant Keys Removed. In *Proceedings of Graphics Interface 2014 (GI '14)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 43–50. <http://dl.acm.org/citation.cfm?id=2619648.2619657>
3. Shiri Azenkot and Shumin Zhai. 2012. Touch Behavior with Different Postures on Soft Smartphone Keyboards. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '12)*. ACM, New York, NY, USA, 251–260. DOI: <http://dx.doi.org/10.1145/2371574.2371612>
4. Joanna Bergstrom-Lehtovirta and Antti Oulasvirta. 2014. Modeling the Functional Area of the Thumb on Mobile Touchscreen Surfaces. In *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1991–2000. DOI: <http://dx.doi.org/10.1145/2556288.2557354>
5. Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1 Euro Filter: A Simple Speed-based Low-pass Filter for Noisy Input in Interactive Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2527–2530. DOI: <http://dx.doi.org/10.1145/2207676.2208639>
6. Steven J. Castellucci and I. Scott MacKenzie. 2011. Gathering Text Entry Metrics on Android Devices. In *CHI '11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11)*. ACM, New York, NY, USA, 1507–1512. DOI: <http://dx.doi.org/10.1145/1979742.1979799>
7. Steven J. Castellucci and I. Scott Mackenzie. 2013. Gestural Text Entry Using Huffman Codes. In *Proceedings of the International Conference on Multimedia and Human Computer Interaction (ASET '13)*. 119–1–119–8.
8. 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. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1499–1508. DOI: <http://dx.doi.org/10.1145/2702123.2702425>
9. Daniel Fitton, I. Scott MacKenzie, Janet C. Read, and Matthew Horton. 2013. Exploring Tilt-based Text Input for Mobile Devices with Teenagers. In *Proceedings of the 27th International BCS Human Computer Interaction Conference (BCS-HCI '13)*. British Computer Society, Swinton, UK, UK, Article 25, 6 pages. <http://dl.acm.org/citation.cfm?id=2578048.2578082>
10. 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. DOI: <http://dx.doi.org/10.1145/2858036.2858242>
11. Yves Guiard. 1987. Asymmetric Division of Labor in Human Skilled Bimanual Action. *Journal of Motor Behavior* 19, 4 (1987), 486–517. DOI: <http://dx.doi.org/10.1080/00222895.1987.10735426> PMID: 15136274.
12. Ken Hinckley and Hyunyoung Song. 2011. Sensor Synaesthesia: Touch in Motion, and Motion in Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 801–810. DOI: <http://dx.doi.org/10.1145/1978942.1979059>
13. Christian Holz and Patrick Baudisch. 2010. The Generalized Perceived Input Point Model and How to Double Touch Accuracy by Extracting Fingerprints. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 581–590. DOI: <http://dx.doi.org/10.1145/1753326.1753413>
14. Christian Holz and Patrick Baudisch. 2011. Understanding Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 2501–2510. DOI: <http://dx.doi.org/10.1145/1978942.1979308>
15. Steven Hoober. 2013. How Do Users Really Hold Mobile Devices? (2013). <http://www.uxmatters.com/mt/archives/2013/02/how-do-users-really-hold-mobile-devices.php>.
16. Eleanor Jones, Jason Alexander, Andreas Andreou, Pourang Irani, and Sriram Subramanian. 2010. GesText: Accelerometer-based Gestural Text-entry Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10)*. ACM, New York, NY, USA, 2173–2182. DOI: <http://dx.doi.org/10.1145/1753326.1753655>
17. Per Ola Kristensson. 2007. *Discrete and Continuous Shape Writing for Text Entry and Control*. Ph.D. Dissertation. Institute of Technology, Linköping University, Sweden.
18. Per Ola Kristensson and Keith Vertanen. 2011. Asynchronous Multimodal Text Entry Using Speech and Gesture Keyboards.. In *Annual Conference of the International Speech Communication Association (Interspeech)*. 581–584. http://www.isca-speech.org/archive/interspeech_2011/i11_0581.html

19. Per Ola Kristensson and Shumin Zhai. 2004. SHARK²: A Large Vocabulary Shorthand Writing System for Pen-based Computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, NY, USA, 43–52. DOI: <http://dx.doi.org/10.1145/1029632.1029640>
20. I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. ACM, New York, NY, USA, 754–755. DOI: <http://dx.doi.org/10.1145/765891.765971>
21. Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. 2014. Vulture: A Mid-air Word-gesture Keyboard. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1073–1082. DOI: <http://dx.doi.org/10.1145/2556288.2556964>
22. Alexander Ng, Stephen A. Brewster, and John H. Williamson. 2014. Investigating the Effects of Encumbrance on One- and Two- Handed Interactions with Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1981–1990. DOI: <http://dx.doi.org/10.1145/2556288.2557312>
23. Kurt Partridge, Saurav Chatterjee, Vibha Sazawal, Gaetano Borriello, and Roy Want. 2002. TiltType: Accelerometer-supported Text Entry for Very Small Devices. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology (UIST '02)*. ACM, New York, NY, USA, 201–204. DOI: <http://dx.doi.org/10.1145/571985.572013>
24. Mahfuz Rahman, Sean Gustafson, Pourang Irani, and Sriram Subramanian. 2009. Tilt Techniques: Investigating the Dexterity of Wrist-based Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1943–1952. DOI: <http://dx.doi.org/10.1145/1518701.1518997>
25. Jun Rekimoto. 1996. Tilting Operations for Small Screen Interfaces. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology (UIST '96)*. ACM, New York, NY, USA, 167–168. DOI: <http://dx.doi.org/10.1145/237091.237115>
26. Shyam Rey, Shumin Zhai, and Per Ola Kristensson. 2015. Performance and User Experience of Touchscreen and Gesture Keyboards in a Lab Setting and in the Wild. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 679–688. DOI: <http://dx.doi.org/10.1145/2702123.2702597>
27. Stephen Sale. 2014. OTT messaging volumes. (2014). <http://www.analysismason.com/About-Us/News/Insight/OTT-messaging-volumes-Jan2014-RDMV0/>.
28. Vibha Sazawal, Roy Want, and Gaetano Borriello. 2002. The Unigesture Approach. In *Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction (Mobile HCI '02)*. Springer-Verlag, London, UK, UK, 256–270. <http://dl.acm.org/citation.cfm?id=645739.666599>
29. Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. 2005. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. In *Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction (INTERACT'05)*. Springer-Verlag, Berlin, Heidelberg, 267–280. DOI: http://dx.doi.org/10.1007/11555261_24
30. Swiftkey. 2014. Swiftkey Tilt. (2014). <https://blog.swiftkey.com/swiftkey-tilt/>.
31. Swype. 2014. Swype. (2014). <http://www.swype.com>.
32. Robert J. Teather and I. Scott MacKenzie. 2014. Position vs. Velocity Control for Tilt-based Interaction. In *Proceedings of Graphics Interface 2014 (GI '14)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 51–58. <http://dl.acm.org/citation.cfm?id=2619648.2619658>
33. 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 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2001–2004. DOI: <http://dx.doi.org/10.1145/2556288.2557088>
34. Keith Vertanen and Per Ola Kristensson. 2014. Complementing Text Entry Evaluations with a Composition Task. *ACM Trans. Comput.-Hum. Interact.* 21, 2, Article 8 (Feb. 2014), 33 pages. DOI: <http://dx.doi.org/10.1145/2555691>
35. William S. Walmsley, W. Xavier Snelgrove, and Khai N. Truong. 2014. Disambiguation of Imprecise Input with One-dimensional Rotational Text Entry. *ACM Trans. Comput.-Hum. Interact.* 21, 1, Article 4 (Feb. 2014), 40 pages. DOI: <http://dx.doi.org/10.1145/2542544>
36. Lars Weberg, Torbjorn Brange, and Asa Wendelbo Hansson. 2001. A Piece of Butter on the PDA Display. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems (CHI EA '01)*. ACM, New York, NY, USA, 435–436. DOI: <http://dx.doi.org/10.1145/634067.634320>
37. Daniel Wigdor and Ravin Balakrishnan. 2003. TiltText: Using Tilt for Text Input to Mobile Phones. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST '03)*. ACM, New York, NY, USA, 81–90. DOI: <http://dx.doi.org/10.1145/964696.964705>
38. John Williamson and Roderick Murray-Smith. 2005. Switching and Learning in Feedback Systems. Springer-Verlag, Berlin, Heidelberg, Chapter Hex: Dynamics and Probabilistic Text Entry, 333–342. <http://dl.acm.org/citation.cfm?id=2137592.2137610>

39. Jacob O. Wobbrock and Brad A. Myers. 2006. Analyzing the Input Stream for Character- Level Errors in Unconstrained Text Entry Evaluations. *ACM Trans. Comput.-Hum. Interact.* 13, 4 (Dec. 2006), 458–489. DOI:<http://dx.doi.org/10.1145/1188816.1188819>
40. Shumin Zhai and Per Ola Kristensson. 2003. Shorthand Writing on Stylus Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 97–104. DOI:<http://dx.doi.org/10.1145/642611.642630>
41. Shumin Zhai and Per Ola Kristensson. 2012. The Word-gesture Keyboard: Reimagining Keyboard Interaction. *Commun. ACM* 55, 9 (Sept. 2012), 91–101. DOI:<http://dx.doi.org/10.1145/2330667.2330689>