Text Entry on Tiny QWERTY Soft Keyboards

Luis A. Leiva[1,*] **Alireza Sahami**[2] **Alejandro Catalá**[3] **Niels Henze**[2] **Albrecht Schmidt**[2]
[1]PRHLT Research Center [2]hciLab [3]ISSI-DSIC
[1,3]Universitat Politècnica de València [2]Universität Stuttgart
[1]llt@acm.org [2]{name.surname}@vis.uni-stuttgart.de [3]acatala@dsic.upv.es

ABSTRACT
The advent of wearables (e.g., smartwatches, smartglasses, and digital jewelry) anticipates the need for text entry methods on very small devices. We conduct fundamental research on this topic using 3 qwerty-based soft keyboards for 3 different screen sizes, motivated by the extensive training that users have with qwerty keyboards. In addition to ZoomBoard (a soft keyboard for diminutive screens), we propose a callout-based soft keyboard and ZShift, a novel extension of the Shift pointing technique. We conducted a comprehensive user study followed by extensive analyses on performance, usability, and short-term learning. Our results show that different small screen sizes demand different types of assistance. In general, manufacturers can benefit from these findings by selecting an appropriate qwerty soft keyboard for their devices. Ultimately, this work provides designers, researchers, and practitioners with new understanding of qwerty soft keyboard design space and its scalability for tiny touchscreens.

Author Keywords
Text Entry; Small Screens; Small Devices; QWERTY
ACM Classification Keywords
H.5.2 User Interfaces: Prototyping; Screen design

INTRODUCTION
With the ongoing breakthrough of wearables, such as smartwatches or digital jewelry, text entry on devices with very small screens (1" wide or less) becomes increasingly relevant and a challenging issue, simply because space is at a premium. A number of approaches have been proposed to enter text on such devices. However, today every text entry technique or keyboard layout based on a touchscreen has to compete with qwerty.[1] Users are only willing to switch and use a different keyboard if the technique is easy to use or learn. The low startup speed, at least partially, precludes the success of a number of text entry techniques that offered high-speed performance after intensive training; e.g., [24, 30]. Compared to gesture-based entry techniques [41] or multi-chord keyboards [24], qwerty keyboards have the advantage that users are already familiarized with the layout and the input technique is easy to understand.

Based on the assumption that standard qwerty soft keyboards are impractical on very small screens, Oney *et al.* [29] proposed ZoomBoard, a qwerty-based multi-tap soft keyboard. Initially, a qwerty keyboard is displayed onscreen. When the user taps on the keyboard, it zooms in and shows an enlarged version of the tapped region. The user can then select a character with an additional tap on the enlarged keyboard region. Oney *et al.* conducted a preliminary evaluation on a 16 mm wide (roughly 0.5") keyboard, and observed that ZoomBoard outperforms a same-sized qwerty soft keyboard that provides no additional assistance for the user. However, the inherent need of using multiple taps to enter a single character makes it unlikely that ZoomBoard would perform as well on larger soft keyboards.

A number of commercial devices with small touchscreens are actually wider than 0.5" (or 0.7" diagonal); e.g. the Sony SmartWatch (1.3"), the i'm S.p.A. watch (1.54"), the Samsung Galaxy Gear (1.63"), or the iPod Nano (available at 1.5" and 2.5"). It is therefore unclear up to which size a zooming approach provides an actual benefit and at which point it will be outperformed by a single-tap approach. In addition, even current standard implementations of qwerty soft keyboards provide different types of assistance for the user. Callouts above the key that are displayed when the finger lands on a key, for example, aim to address the occlusion problem ("fat-finger") and increase user performance. Therefore, different types of assistance for small qwerty soft keyboards still remain largely unexplored.

We investigate the scalability of 3 qwerty-based text entry techniques for 3 diminutive screen sizes, by using 16 mm, 21.3 mm, and 28.4 mm wide soft keyboards. In addition to ZoomBoard, we implemented a qwerty soft keyboard that presents the currently selected character above the keyboard, similar to the callouts provided by current smartphone keyboards. We also implemented ZShift, an extension of the Shift pointing technique [39] that we have adapted for text entry. We show that different screen sizes demand different keyboard techniques. For instance, ZoomBoard performs well on the smallest screen size whereas ZShift scales better to larger screen sizes. We found that the three keyboards approach reasonable entry speeds along with competitive accuracy. We also observed that users got quickly familiarized with all keyboards after entering just 5 sentences with each keyboard-screen combination within a single session. In general, manufacturers can benefit from these findings by selecting an appropriate qwerty soft keyboard for their devices. Ultimately, this work provides designers, researchers, and practitioners with new understanding of qwerty soft keyboard design space and its scalability for tiny touchscreens.

[*]Work done while visiting the hciLab in Stuttgart.
[1]We use QWERTY in lowercase form to improve typesetting.

CHI 2015, April 18–23, 2015, Seoul, Republic of Korea.

ACM 978-1-4503-3145-6/15/04 ...$15.00.
http://dx.doi.org/10.1145/2702123.2702388

RELATED WORK

The concept of very small interactive mobile devices has recently sparked interest well beyond HCI research. In particular, wearables such as smartwatches, smartglasses, and digital jewelry are becoming widely available to consumers. Interestingly, these devices can receive notifications in many forms but there is usually no direct way of replying [19].

Speech input seems to be an obvious choice to enter short messages, names, or addresses on very small devices. However, there are situations where it is too noisy or inappropriate to use; e.g., asking for personal data on an overcrowded environment. Researchers have proposed to use handwriting to enter text on mobile devices [17, 45], though it is difficult for the user to see what is currently being written on very small screens. In addition, handwritten text (much like voice) is prone to recognition errors. Alternatively, the rear of the device can be used for interaction [1], though it is typically unavailable on consumer devices. Another possibility are wrist-worn devices like Facet [23], a circular bracelet of multi touch displays, although their form factor is too big to be practical.

Soft Keyboard Layouts

In general, physical qwerty keyboards are commonplace and the first text entry device for most users. Thus, other techniques and keyboard layouts have to compete with it. Even soft keyboard layouts optimized for movement efficiency following Fitts' law and character frequencies such as OPTI [28] or ATOMIK [44] showed that users need to invest non-negligible time until the qwerty layout is eventually outperformed. Most users are not willing to switch to a different input technique or even a different layout if it does not provide a similar startup speed. In fact this is the dominant factor for adoption of text entry techniques [6]. A prominent high-performance example is Twiddler, a one-handed chording keyboard [24] that allows users to achieve up to 60 WPM—comparable to a physical qwerty keyboard [34], but only after months of training.

Currently, qwerty-based text entry predates other alternatives. Due to the proliferation of touchscreen devices, different approaches have been developed to improve qwerty soft keyboards, from subtle changes to the internal processing of touch input [10, 12] to slight changes of the button layout [4]. Himberg *et al.* [13] developed an adaptive numerical soft keyboard that observes where the user is touching and adapt the shape of the virtual keys to reduce the error rate. Similar work by Kristensson and Zhai [20] uses geometric pattern matching to reduce the error rate for stylus-based text entry. Gunawardana *et al.* [10] developed an anchored keyboard adaptation, and a user-simulated study suggested that it may reduce the error rate. Using these techniques on small devices might not be practical due to the number of keys involved and the fat-finger problem. While previous works have explored multi-tap and predictive alternatives [5, 14, 16, 19], researchers still tried to shrink down a qwerty keyboard to fit on very small touchscreen. For instance, Kim *et al.* [18] used one key for interaction, Minuum[2] compressed the qwerty layout to one line, and Oney *et al.* [29] used iterative zooming to enlarge the keys.

[2] `http://minuum.com`

Interaction Techniques for Text Entry

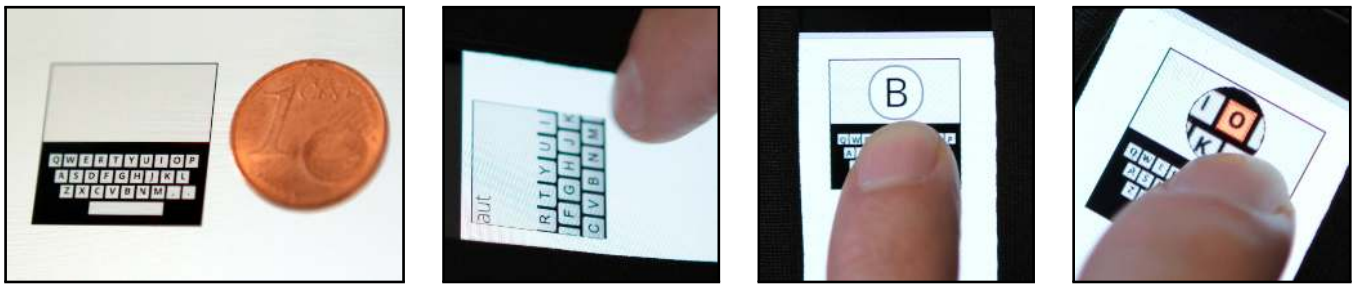
Several interaction techniques using different sensors show promise for entering text on very small devices; e.g., using magnetometers [11], tilting a wrist-worn device [31, 36], or combining physical pan, twist, tilt, and clicks [42]. Such techniques remain to be further explored, but also remain difficult to deploy in practice.

Gesture-based text entry techniques such as EdgeWrite [41] or Quikwriting [32] became common on mobile devices that required a stylus as input. Reducing the input space to well-delimited zones simplified recognition accuracy, which was an issue in former approaches such as Graffiti [27] or Unistroke [9]. Other works pursued a minimal set of 4 keys, or interaction zones if used in gesture-based systems, that would allow efficient text entry such as MDITIM [15], LURD [7], and H4-Writer [26]. The key problem of these gesture-based input techniques is that an additional stylus is required, and thus yet another device that might be even larger than the actual device the user interacts with. On the other hand, using the finger in lieu of a stylus leads to the fat-finger problem that is particularly severe on very small screens.

Small Target Acquisition Techniques

Another strand of research focuses on techniques to select small targets with a finger without changing the size of the target while achieving an acceptable error rate. In Shift [39] target selection is approached through callouts showing a copy of the area occluded by the finger in a non-occluded area. In TapTap [35] the occluded area is magnified and the user has to touch the desired target with a second touch, similar to ZoomBoard. In Escape [43] targets are visually enhanced with arrows that indicate the direction in which the user has to drag the finger after touching a target. Those interaction techniques are not very well suited for text entry since additional interactions are required, which in turn require more time and higher mental effort compared with a simple touch. Nevertheless they can play a relevant role if they are properly combined in new scenarios on small touchscreens, where target selection can be specially challenging and special needs are therefore required [33]. For instance, Swipeboard [3] allows users to enter text with two swipes: the first swipe specifies the region where the character is located, and the second swipe specifies the character within that region. This approach is target-agnostic and so after some training users can perform eyes-free, shorthand input.

In sum, a significant body of research has investigated text entry for small devices. While diverse alternatives to qwerty-based text entry have been proposed, the comparatively high usability of qwerty keyboards suggests that these keyboards will play a major role for small touchscreen-capable devices that currently hit the market. ZoomBoard is the most recent approach that has been especifically designed to address this issue on diminutive screens. For larger screen sizes, however, callout-based techniques can be used instead. Furthermore, it remains unclear how these approaches actually perform in comparison. In particular, it is unclear how each technique scales and which technique is appropriate for which device size. Our work is the first to address this choice.



(a) Reference scale (b) ZoomBoard (c) Callout (d) ZShift

Figure 1: Our three prototypes. As a reference, a 1 cent Euro coin (16 mm, or 0.6", 3 mm smaller than a US penny) is shown in (a).

THREE TINY QWERTY SOFT KEYBOARDS

To date, ZoomBoard is the only qwerty soft keyboard that has been specifically designed to enter text on tiny touchscreens. However, we propose two alternatives to this technique, motivated by the following considerations:

1. A qwerty keyboard layout, due to its 2:1 (or higher) aspect ratio, typically takes up only half of the screen space. Therefore, the remaining space can be available to display information related to text entry.
2. Tiny touchscreen sizes may range from "very small" (less than 1") to "moderately large" (e.g., 2.5" for the newest iPod Nano). Thus, different text entry techniques may perform differently depending on the available space onscreen.

Figure 1 shows the three prototypes we have studied. All prototypes are web-based and have been tested on different browsers, including mobile and desktop computers. The prototypes are released as open source software, so that anyone can contribute to improving them or build alternatives by reusing parts of the code.[3]

ZoomBoard Keyboard

To increase the accuracy with which a key can be acquired, instead of immediate selection, the keyboard zooms in (Figure 1b). Specifically, when the user taps on a key, the keyboard iteratively zooms in (visual magnification) until reaching a certain level of zoom. Then, the user can enter a character with an additional tap. Afterward, the keyboard goes back to the initial zoom level. As the keyboard layout is visible to the user after each tap, less typing errors are likely to occur compared to a non-zooming qwerty soft keyboard, which may suffer from severe occlusion problems on tiny screens.

Callout Keyboard

The Callout keyboard is inspired by the soft keyboards used on current smartphones. When the user touches a key, a callout showing the character that is about to be entered is created in a non-occluded location (the upper part of the screen, Figure 1c). The user can refine the key to be entered by slightly moving the finger on the keyboard, and then enter the character by lifting up the finger. This technique allows the user to enter one character per tap, which might be more efficient than ZoomBoard.

[3] `http://personales.upv.es/luileito/tinyqwerty/`

ZShift Keyboard

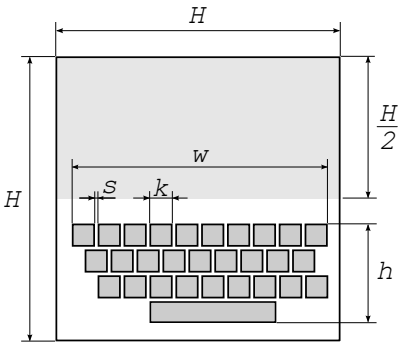
The Callout keyboard has the drawback that once the finger has landed on the touchscreen, it occludes most (if not all) of the keyboard. Therefore, if the user wants to refine key selection, she must rely on her spatial memory to know how keys are exactly arranged. Thus, we provide the user with a stronger hint about where each key is located. We applied the Shift pointing technique [39], which was designed to ease target acquisition but we have extended for text entry. Shift creates a callout showing a copy of the occluded screen area (motor magnification). Using this visual feedback, users might be more accurate while entering text. However, for small keyboards we believe that Shift alone is not sufficient. Thus, we enhance the callout area with one level of zoom over the occluded area, providing also visual feedback of the touched key (Figure 1d), yielding a *Zoomed* Shift technique (ZShift).

Common Features

Following previous keyboard designs that already used swipe gestures to replace touchscreens buttons [8, 22, 29], we apply these gestures to the following functions. On each keyboard prototype, the user can enter a space either by tapping on the space bar or by swiping to the right over the keyboard. To delete text, the user must swipe to the left. To load different keyboard layouts (e.g., one for symbols and numbers, other for punctuation or currency symbols, etc.) the user can swipe either up or down, following a carousel metaphor, allowing thus a continuous, circular navigation through all available keyboard layouts. To submit the entered text, the user must tap on the upper part of the screen (see Figure 2).

EVALUATION

We conducted a controlled user study to compare the three keyboard alternatives using text-copy tasks, as usual in text entry experiments. We tested the 3 keyboards with 3 different sizes (see Figure 2), 9 conditions in total. We simulated a smartwatch using a touch-capable smartphone, in order to eliminate a potential evaluation bias. It must be noted that using actual smartwatches would require a different model for each screen size, resulting in different form factors, touch responsiveness, and screen resolutions. Instead, using the same device for all participants eliminates these undesirable effects. Our evaluation is thus general enough so as to illustrate how text entry would perform on wearables featuring tiny qwerty soft keyboards.



Screen	H	w	h	k	s
Small	18	16.0	6.5	1.5	0.3
Medium	24	21.3	8.6	2.0	0.4
Large	32	28.4	11.4	2.6	0.5

Figure 2: Keyboard definition and the values used for each of the screen sizes. All dimensions are given in millimeters.

Apparatus

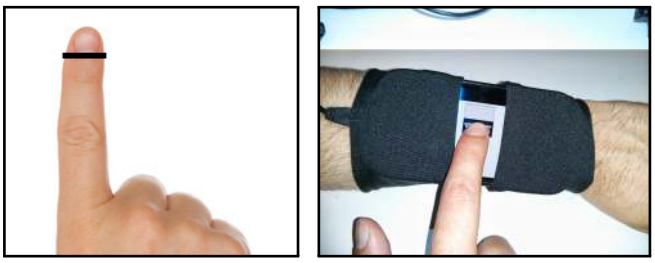
We used a Samsung Nexus S mobile phone running Android 4.1 with a 4" display (233 dpi). The phone was attached in landscape orientation to the non-dominant arm with two wide black strips; see Figure 3b. The strips, in addition to fasten the phone, simulate the edges of a watch and cover the screen in such a way that only the simulated screen width is visible to the user.

The layout of the tested keyboard prototypes are accurately defined in Figure 2, each one being one third larger than its predecessor. It can observed that "small" (18 mm), "medium" (24 mm), and "large" (32 mm) screen sizes are actually all very small compared to current smartphone keyboards. Since our prototypes were written in JavaScript, we used the Firefox web browser for Android. The browser was launched in fullscreen mode, so the ribbon at the top of the browser, which also includes the URL box, was not visible to the participants. Similar to the study conducted by Oney *et al.* [29], ZoomBoard was configured to work with one level of animated zoom, so that each character was entered with 2 taps.

Design

We considered two independent factors: *Keyboard* method (3 levels: ZoomBoard, Callout, ZShift) and *Screen* size (3 levels: small, medium, large). We further used 8 dependent variables: 6 performance-related (described in Analysis of Text Entry Performance) and 2 usability-related factors (described in Analysis of Usability and Workload). We also investigated how each condition performed at the phrase level, to obtain a holistic overview about the different keyboard layouts.

We used a repeated measures within-subjects design, i.e., participants were assigned to all treatment levels of every factor combination. The Latin square design was adopted to counterbalance the order of the conditions, i.e., we generated a 9x9 assignments matrix were every single condition followed every other condition only once [2], and each participant followed one of the rows of the assignments matrix. The data



(a) (b)

Figure 3: Measuring index finger's width (3a, black strip) and detail of the evaluation setup (3b).

were analyzed using a two-way multivariate analysis of variance (MANOVA), since there is more than one dependent variable as main outcome, followed by a series of ANOVAs and post-hoc comparisons per each screen size group, where applicable.

Participants

We recruited 20 participants (5 female) aged 21–29 (M=24.7, SD=2.2) using our University's mailing lists. We intentionally wanted a rather broad sample and recruited participants with many different backgrounds; e.g. Mechanical Engineering, Informatics, or Physics. All participants regularly used PC keyboards. Thirteen participants stated that they could perform blind-typing on a PC keyboard. Seventeen participants were right-handed and 17 owned touchscreen smartphones. Each participant was paid 10 € at the end of the evaluation.

Procedure

We conducted the study in a calm office environment. Participants were seated during the whole study, as we anticipated that it would take about one hour per participant. Each participant was briefly described the purpose of the study to begin with. We measured the width of their dominant hand's index finger with a digital caliper, for which it was aligned with the distal interphalangeal joint (see Figure 3a). The average size of the index finger was 16.1 mm (SD=1.4). This gives an approximate idea of how much of a very small screen is occluded by the finger.

Participants started the study by signing in a consent form and answering a demographics questionnaire. Next, the phone was attached to the non-dominant arm. Through a short guided demo, the three keyboards were presented and explained to each participant. People were asked to type their full name using each keyboard design on the medium-size keyboard. They were told to use the index finger of the dominant hand for entering text during the whole study. This warm-up session took approximately 1–2 minutes on average per condition. Afterward, the actual evaluation began.

Each participant had to enter 5 phrases for each of the 9 different keyboard-screen combinations, resulting in 45 phrases submitted per participant, 900 phrases in total. As previously commented, we used a Latin square design to counterbalance the order of the conditions. This procedure reduces learning effects as well as asymmetrical skill transfer across conditions.

Phrases were picked at random from the MacKenzie and Soukoreff phrase set [25], which is a well-known standard dataset to conduct text entry experiments. All phrases had neither punctuation symbols or numbers, and were lowercased in order to let participants easily focus on each keyboard technique. A phrase was shown at a time above each keyboard, and we ensured that all phrases were different for each participant and condition; in fact no phrase was entered twice in any of the conditions. Participants were asked to enter the presented text as quickly and accurately as possible. They were allowed to correct mistakes as they went, for which they would use the left swipe gesture to delete the last character. Each phrase was permanently shown to the participants until they submitted it, in order to avoid memorability bias.

Participants were able to practice and get accustomed to the keyboard-screen combination used in each condition before actually evaluating it. These attempts took between 2 and 5 minutes per participant. After finishing typing a phrase, participants had to tap on the upper part of the soft keyboard to submit the phrase and load a new one. When each condition finished, participants were asked to answer the SUS and NASA-TLX questionnaires on a nearby desktop computer.

RESULTS

In our prototypes, for pragmatic reasons, a phrase was submitted by tapping on the upper part of the screen. It turned out that in 12 cases participants accidentally tapped on that part when tried to reach a key from the first row of the keyboards. To remove the accidentally submitted phrases, we only considered those phrases that were transcribed at least by 50%. This resulted in 888 phrases for analysis, which anecdotally correspond to 12.3 hours of typing data.

A MANOVA test was first performed to take into account the interaction effects between variables and protect against inflating the Type 1 error in follow-up ANOVAs and post-hoc comparisons, whether appropriate. Prior to conducting the MANOVA, a series of Pearson correlations were performed in order to test the MANOVA assumption that the dependent variables would be correlated with each other. Table 1 summarizes these correlations. A non-significant result after the Box's M test ($p > .05$) indicated a lack of evidence that the homogeneity of variance-covariance matrix assumption was violated. No univariate or multivariate outliers were evident and MANOVA was considered thus to be an appropriate analysis technique.

	KSPC	WPM	CER	Nerr	Cerr	Ceff	SUS	TLX
KSPC	—							
WPM	**-0.21**	—						
CER	0.08	**-0.14**	—					
Nerr	0.05	-0.06	**0.95**	—				
Cerr	**0.33**	**-0.35**	0.06	-0.06	—			
Ceff	**0.19**	-0.10	0.07	0.01	**0.06**	—		
SUS	**-0.04**	**0.39**	**-0.15**	**-0.09**	**-0.03**	**-0.20**	—	
TLX	**0.06**	**-0.36**	**0.24**	**0.15**	**0.02**	**0.17**	**-0.04**	—

Table 1: Pearson's r correlation between all dependent variables. Statistical significance ($p < .05$) is denoted in bold typeface.

MANOVA tested the hypothesis that there was one or multiple differences of the mean between *Keyboard* levels (ZoomBoard, Callout, ZShift) and screen *Size* levels (small, medium, large). Significant multivariate effects were found among the 9 conditions, both regarding *Keyboard* [$F_{2,171} = 23.69$, $p < .0001, \eta_p^2 = 0.53$] and *Size* [$F_{2,171} = 8.49$, $p < .0001, \eta_p^2 = 0.29$]. In addition, a significant *Keyboard*Size* interaction was found [$F_{4,171} = 1.55$, $p = .028, \eta_p^2 = 0.13$]. We therefore split the dataset by screen size and performed univariate ANOVAs, with appropriately adjusted significance levels to guard against the risk of over-testing the data. All comparisons used the Holm-Bonferroni correction.

The following analysis includes four parts. First, we investigate text entry performance using all keyboard-screen combinations. Next, we assess usability and workload through the analysis of the SUS and NASA-TLX questionnaires. We also provide anecdotal evidence of the typing errors commited by the participants. Finally, we assess user's short-term learning on a per-trial basis.

Analysis of Text Entry Performance

We assesed text entry performance using the following measures. Certainly there are more conceivable measures that could be used, but for brevity's sake we report the most relevant and well-established measures in the literature.

Analysis of Words Per Minute and Key Stroke Per Character

Words Per Minute (WPM) and Key Stroke Per Character (KSPC) are widely used measures of input speed. For standardization purposes, in WPM a word is defined as five consecutively entered characters, including spaces. KSPC is the number of interactions (e.g., taps, swipes) required to enter a character, including backspaces. KSPC is device-dependent, and thus ZoomBoard has a theoretical lower bound of 2.0, though this can be lowered down to 1.84 if the swipe gesture is used for entering spaces [29].

	ANOVA			Keyboard		
Screen	$F_{2,57}$	p-value	η_p^2	ZoomBoard	Callout	ZShift
Small	6.89	.002	0.19	6.0 (1.4)	4.3 (1.7)	5.4 (1.2)
Medium	0.70	.498	0.02	7.8 (1.2)	7.1 (2.0)	7.2 (2.3)
Large	0.96	.386	0.03	8.2 (1.2)	8.3 (2.3)	9.1 (2.9)

Table 2: WPM results (higher is better). Mean values are shown in the Keyboard column. SDs are denoted in parentheses.

As shown in Table 2, WPM differences were found to be statistically significant only for the small screen. Post-hoc pairwise comparisons using the t-test (Holm-Bonferroni corrected) revealed that the Callout keyboard performed worse than the other alternatives.

As shown in Table 3, KSPC differences were found to be statistically significant for all screen sizes. Post-hoc pairwise comparisons using the t-test (Holm-Bonferroni corrected) revealed that ZoomBoard required more KSPC than the other alternatives for all screen sizes. ZShift was the best performer overall. For medium and large sizes, there were no significant differences between ZShift and Callout.

	ANOVA			Keyboard		
Screen	$F_{2,57}$	p-value	η_p^2	ZoomBoard	Callout	ZShift
Small	36.88	<.0001	0.56	2.7 (0.5)	1.8 (0.4)	1.5 (0.2)
Medium	50.11	<.0001	0.63	2.2 (0.2)	1.5 (0.2)	1.4 (0.2)
Large	112.68	<.0001	0.79	2.1 (0.1)	1.4 (0.1)	1.3 (0.2)

Table 3: KSPC results (lower is better). Mean values are shown in the Keyboard column. SDs are denoted in parentheses.

Analysis of Character Error Rate

Character Error Rate (CER) is the most widely used measure of accuracy. CER is computed as the Damerau-Levenshtein distance between the submitted text and the reference text, normalized by the number of characters in the reference text.

	ANOVA			Keyboard		
Screen	$F_{2,57}$	p-value	η_p^2	ZoomBoard	Callout	ZShift
Small	4.41	.016	0.13	1.1 (1.3)	2.6 (2.1)	1.3 (1.3)
Medium	0.51	.603	0.01	1.2 (2.1)	0.8 (1.0)	1.3 (1.8)
Large	0.97	.383	0.03	1.4 (2.3)	0.7 (0.9)	0.9 (1.3)

Table 4: CER results (lower is better). Mean values are shown in the Keyboard column. SDs are denoted in parentheses.

As shown in Table 4, CER differences were found to be statistically significant for the small screen size. Post-hoc pairwise comparisons using the t-test (Holm-Bonferroni corrected) revealed that Callout performed worse than the other alternatives. No significant differences between ZoomBoard and ZShift were found.

Analysis of Device-independent Performance Measures

CER is an incomplete accuracy measure, in the sense that it cannot differentiate among the errors that were *corrected* by the user, yet these are an important aspect of text entry. Also, KSPC does not differentiate between the cost of committing errors and the cost to fix them. For these reasons, here we use measures that consider the *input stream* [37]. The input stream contains information about all text that has been entered and erased. Thus it is both device- and method-independent [40].

Corrected Error Rate (Cerr) provides an equivalent for the KSPC (the cost of typing and fixing errors). Some authors suggest to break down Cerr in 2 sub-metrics [38], but this fine-grained distinction provides little insights when there is no "pathologic error correction",[4] as in our data. Second, Non-corrected Error Rate (Nerr) is analogous to CER (a measure of the errors remaining in the transcribed text). Nerr and Cerr together account for the Total error rate (Terr), which will not be reported here for brevity's sake. Finally, Correction Efficiency (Ceff) measures the ease with which the user performed error corrections.

Differences were found to be statistically significant in terms of Cerr for medium and large screen sizes (Table 5), in terms of Nerr for small screen size (Table 6), and in terms of Ceff for medium screen size (Table 7). Post-hoc pairwise comparisons using the t-test (Holm-Bonferroni corrected) revealed statistical significance.

[4]The user notices he has committed an error and goes backspacing destructively to the error and re-enters the correct text.

	ANOVA			Keyboard		
Screen	$F_{2,57}$	p-value	η_p^2	ZoomBoard	Callout	ZShift
Small	1.63	.203	0.05	14.2 (6.2)	17.3 (5.9)	14.1 (6.7)
Medium	10.77	.0001	0.37	6.8 (4.8)	14.0 (5.1)	12.6 (5.5)
Large	12.32	.0001	0.43	5.9 (3.2)	11.7 (3.4)	11.4 (5.4)

Table 5: Cerr results (lower is better). Mean values are shown in the Keyboard column. SDs are denoted in parentheses.

	ANOVA			Keyboard		
Screen	$F_{2,57}$	p-value	η_p^2	ZoomBoard	Callout	ZShift
Small	3.48	.037	0.10	0.8 (1.0)	1.8 (1.6)	0.9 (1.1)
Medium	0.72	.490	0.02	1.0 (1.7)	0.5 (0.6)	0.9 (1.1)
Large	1.91	.157	0.06	1.5 (2.7)	0.5 (0.6)	0.7 (1.0)

Table 6: Nerr results (lower is better). Mean values are shown in the Keyboard column. SDs are denoted in parentheses.

	ANOVA			Keyboard		
Screen	$F_{2,57}$	p-value	η_p^2	ZoomBoard	Callout	ZShift
Small	1.05	.356	0.03	93.0 (13.4)	92.7 (13.7)	85.7 (24.4)
Medium	6.11	.003	0.17	72.2 (32.9)	94.0 (14.6)	93.0 (13.4)
Large	1.61	.207	0.05	78.0 (19.3)	88.0 (16.4)	85.0 (18.2)

Table 7: Ceff results (higher is better). Mean values are shown in the Keyboard column. SDs are denoted in parentheses.

Analysis of Usability and Workload

We measured usability through the System Usability Scale (SUS) and perceived workload through the NASA Task Load Index (TLX). SUS comprises 10 questions assessed in a 1–5 Likert scale, giving a global view of subjective assessments of usability. The overall SUS score ranges from 0 to 100 (the higher the better). TLX comprises 6 questions assessed in a 1–10 scale, each question related to one of the following dimensions: mental demand, physical demand, temporal demand, performance, effort, and frustration. The overall TLX score ranges from 0 to 100 (the lower the better).

Differences in SUS (Table 8) and TLX (Table 9) were found to be statistically significant for the small size. This was also true regarding TLX differences for the medium size. All other ANOVAs did not reveal statistically significant differences.

	ANOVA			Keyboard		
Screen	$F_{2,57}$	p-value	η_p^2	ZoomBoard	Callout	ZShift
Small	6.25	.003	0.17	67.8 (13.1)	51.1 (11.5)	59.7 (19.1)
Medium	1.22	.300	0.04	75.2 (14.4)	68.1 (15.0)	73.8 (16.2)
Large	2.56	.087	0.08	83.5 (11.4)	77.1 (15.2)	85.6 (9.8)

Table 8: SUS results (higher is better). Mean values are shown in the Keyboard column. SDs are denoted in parentheses.

Post-hoc comparisons (Holm-Bonferroni corrected) of SUS and TLX for the small size revealed that the Callout keyboard was perceived as significantly worse than the other alternatives,

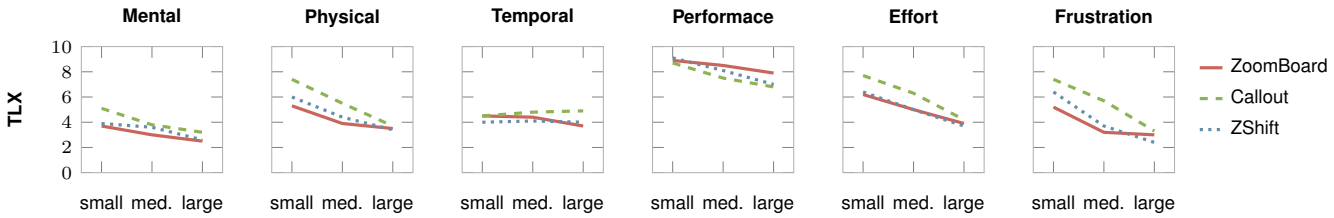


Figure 4: Scores for each of the 6 TLX dimensions (lower is better). Performance measures "how difficult was to enter text."

	ANOVA			Keyboard		
Screen	$F_{2,57}$	p-value	η_p^2	ZoomBoard	Callout	ZShift
Small	4.91	.010	0.14	21.9 (4.2)	26.0 (4.3)	22.5 (4.8)
Medium	3.97	.024	0.12	18.8 (4.4)	22.5 (5.3)	19.4 (3.7)
Large	1.24	.294	0.04	17.2 (3.9)	18.7 (4.4)	16.9 (3.6)

Table 9: TLX results (lower is better). Mean values are shown in the Keyboard column. SDs are denoted in parentheses.

and ZoomBoard was perceived better in terms of SUS than ZShift but not in terms of TLX. Regarding TLX for medium screen size, the Callout keyboard scored significantly higher than ZoomBoard and ZShift, and no statistically significant differences were found between ZoomBoard and ZShift. Since all other ANOVAs did not reveal statistically significant differences between conditions in the remaining screen sizes (both for SUS and TLX), no further post-hoc tests were performed.

Analysis of Typing Errors

We observed that 160 phrases had some transcription errors, either because of typos (*Typo*, 110 cases), because a different word—although grammatically correct—was entered (*Diff*, 69), or because fewer words (*Less*, 31) or more words (*More*, 15) than the reference phrase were entered. Figure 5 shows a histogram of the error distribution. As observed, most of the submitted phrases were error-free, and at most 4 phrases had 3 transcription errors. Table 10 shows some examples of the errors committed by the users.

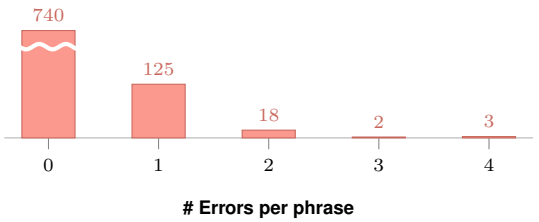


Figure 5: Histogram of phrases containing transcription errors. Most of the submitted phrases were error-free.

Analysis of Short-term Learning

We analyzed the collected data to determine improvements (or the lack thereof) throughout each condition. The measures were averaged on a per-trials basis, so that there were 5 observations or "data points" for each of the 9 tested conditions.

As shown in Figure 6, participants performed consistently better while entering the fifth phrase in all conditions. In addition, we observed that performance improved as screen size increased. This was so for all measures, though we report results about WPM and KSPC by way of illustrative examples. A simple linear regression analysis for each condition revealed that all fits were statistically significant ($R^2 > 0.8, p < .001$).

Error type	Example (reference text above transcribed phrase)
Typo	`an occasional taste of chocolate` `an occasional taste of chpcolate`
Diff	`this mission statement is baloney` `the mission statement is baloney`
Less	`the back yard of our house` `the backyard of our house`
More	`i hate baking pies` `i hate baking and pies`

Table 10: Examples of transcription errors (underlined).

DISCUSSION

This work provides valuable new knowledge for text entry researchers, and for interaction designers that want to incorporate text entry methods on very small screens. In the following we comment on the main findings derived from our study.

Findings

Overall, participants liked ZoomBoard and ZShift the most. A frequent observation was that, on medium and large screens, participants complained about having to issue 2 taps with ZoomBoard to enter each character. One participant stated that *"ZoomBoard was especially irritating on the larger screen"*, who found ZShift to be the most likable alternative.

For the small screen (16 mm), our results show that ZoomBoard provides the highest WPM (6.04) followed by ZShift (5.41), though this difference is not statistically significant. Further, ZoomBoard's KSPC is almost twice higher than ZShift, this difference being statistically significant. On the other hand, there is no significant CER differences between ZoomBoard and ZShift. This suggests that both ZoomBoard and ZShift are less error-prone than Callout on the small screen. It was interesting to notice that all entry speeds are very slow compared with current smartphones, where WPM is typically about five times higher. Furthermore, Callout was significantly slower than the other keyboards, especially on the small screen. This suggests that current soft keyboards are not appropriate for use on very small screens, and that other alternatives should be devised, such as ZoomBoard or ZShift.

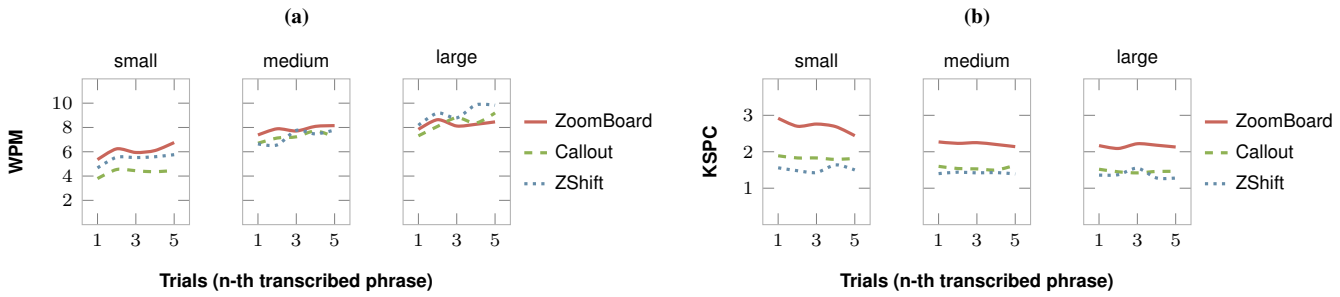


Figure 6: Evolution of performance in terms of WPM (a) and KSPC (b) for all combinations of screen size and keyboard.

For the medium (24 mm) and large (32 mm) screens, no statistically significant differences were found between the keyboards both in terms of WPM and CER, but it was so in terms of KSPC. Thus, all keyboards are usable on the medium and large screens. Using ZShift indeed leads to the lowest KSPC. For the medium screen, however, using the Callout technique requires a higher mental workload and its usability was perceived to be lower than the other techniques. On the other hand, mental workload was not found to be significantly different on the larger screen. Specifically, ZShift usability was perceived to be higher than the other keyboards.

The analysis of TLX dimensions suggests that users are geared toward text entry techniques that do just require 1-step interaction, without lifting the finger from the touchscreen. For instance, temporal demand, effort, and frustration scores were the lowest in ZShift for the large screen, and Callout performed the best for the medium screen. Therefore, having to issue two taps to enter one character seems to be questionable for medium and large screens. In fact, the qualitative comments of our participants indeed reveal the inconvenience of issuing two taps to enter a single character. This was especially mentioned for the larger screen.

Implications for Design

Our findings and related results provide estimates of the advantages and drawbacks attributed to different soft keyboard design choices. Thus, where to use any of the approaches we have studied can be an informed choice depending on the available space onscreen. In the following we summarize the main implications derived from our findings.

The larger the tiny screen, the better

We observed that performance improved as screen size increased. Of special importance is the case of the larger screen, where Callout and ZShift achieved 10 WPM. Interestingly, it was observed that all participants systematically perceived the three keyboards being more usable as screen size increased (Table 9). A detailed analysis of each TLX dimension revealed that this behavior was surprisingly consistent for all keyboard-screen combinations (Figure 4). This puts forward the fact that a few millimeters can make a difference.

Text entry on tiny qwerty keyboards is feasible

The short-term learning analysis suggests that the tested keyboards provide reasonable entry speeds on very small screens along with competitive accuracy. Interestingly, in terms of KSPC, both Callout and ZShift keyboards stabilized around 1.5 on all screen sizes, which means that transcription errors were systematically committed from time to time (Figure 6b). Meanwhile, ZoomBoard approached its innate value of 2.0 when entering the fifth phrase on all conditions. This suggests that our participants were more accurate with ZoomBoard.

Text entry on tiny qwerty keyboards is not so error-prone

Participants were told to type as fast and accurate as possible, and were able to correct errors as they went. Unsurprisingly, the smallest screen was the most error-prone overall, where ZoomBoard and ZShift outperformed Callout. This is so because they provide mechanisms based on zooming and visual context that allows users to deal with the inaccuracy of finger-based input. With Callout, users can still move their finger before lifting it up, however we found that it was not enough to achieve competitive accuracy on diminutive screens. For the larger screen these differences faded away. In the end, we observed that 20% of the submitted phrases had one or more transcription errors. Most of them were typos, therefore incorporating error correcting mechanisms would be plausible and very useful to improve all keyboard alternatives.

Submitting text on tiny screens can be tricky

We observed that simply tapping on the upper part of the screen to submit text can become a trouble, as users could accidentally submit when actually trying to enter a character. Hence, when designing a "submit text" interaction for tiny touchscreens, a more elaborated interaction should be considered; e.g., by issuing a double tap.

Cognitive aspects of text entry on tiny qwerty keyboards

Callout and ZShift require the user to relocate the focus of attention; i.e., she has to look first in which key her finger has landed, then she has to move eyesight to the upper part of the screen to verify it, as the finger is occluding most of the keyboard. Conversely, participants got immediately familiarized with ZoomBoard, since tap+zoom interactions are simple to perform, easy to understand, and do not occlude the keyboard because of the first tap. Therefore, the other techniques are cognitively inferior to begin with and may require a little practice for the very first time. Nevertheless our data shows that ZShift performs similar to ZoomBoard on small and medium screens, and that both ZShift and Callout outperform ZoomBoard on the larger screen.

One qwerty keyboard for one tiny screen

ZoomBoard was evaluated with a fairly small user sample (4 females, 2 males). We can now corroborate that Oney *et al.*'s findings still hold in a more general setting with a broader user sample. In sum, ZoomBoard is a reliable solution for entering text on diminutive screens. Considering that it was outperformed by ZShift on the larger screen, we would probably recommend ZoomBoard for really small devices such as necklaces or earrings, and ZShift for mid-sized devices such as smartwatches or bracelets.

Limitations

Apart from the "usual suspects" in lab-based studies (number of participants, age, etc.), we should mention that we used a relatively small number of phrases for each condition, and they were all different in each condition, for all participants. This may have made true differences between conditions difficult to detect. Nevertheless, all phrases have similar complexity, are moderate in length, easy to remember, and representative of written English [25]. Moreover, each participant tested each keyboard-screen combination for around 10 minutes, so special consideration was put into balancing user effort and data granularity. We hypothesize that a longitudinal between-subjects learnability study (i.e., only 1 keyboard-screen condition per participant instead of 9 conditions) would clearly reveal the actual benefit of using each keyboard over time, because carryover and fatigue effects would had little influence.

Modern soft keyboards use word prediction and error autocorrection mechanisms, and for tiny screens it is plausible that these would be very useful. However, we did not test this effect because it would have introduced another factor that would have compromised the internal validity of the experiment. This actually brings us to the following discussion.

Our study was conducted in a controlled environment, with users performing copy-text tasks. Although it may seem more natural to have users enter free text on an actual wearable device and increase thus the external validity of the experiment, it is critical to make the text entry method the only independent variable in the experiment, and increase thus its internal validity [21]. Indeed, if users were just asked to type "anything as fast and accurately as possible" they would introduce rather biased text. Therefore, we are confident that the observed effects between keyboards and screen sizes are, to a large extent, due to the test conditions.

CONCLUSION AND FUTURE WORK

We investigated the scalability of 3 qwerty-based soft keyboards for 3 small screen sizes. Text entry on tiny touchscreen devices might typically be limited to simple tasks such as writing names, addresses, calendar events, or short messages. However, until now text entry alternatives for such devices were scarce. Our research is based on the existing familiarity that computer users have with qwerty keyboards, a well-established property that every soft keyboard has to compete with, for better or worse.

Through a comprehensive study, we showed that different screen sizes demand different types of assistance. Our results reveal that users subjectively perceive ZoomBoard to be the best choice for the smallest screen (16 mm wide). However, ZShift achieves similar performance—all metrics are statistically comparable, excepting KSPC that is innately higher for ZoomBoard. For the other screen sizes (24 and 32 mm wide) ZShift is the best performer overall. It was interesting to observe that users got quickly familiarized with all text entry techniques.

Overall, the three keyboards we studied approach reasonable entry speeds along with competitive accuracy for entering text on tiny touchscreens. Broadly speaking, a technique like ZoomBoard, which relies on tap+zoom interactions and deals with occlusion by forcing users to lift up their fingers, is competitive for diminutive touchscreens. Nevertheless, more advanced approaches like ZShift, that provide visual feedback through a zoomed callout, may achieve better performance.

Finally, we believe there are 3 research avenues worth following for future work. First, to explore if there are any gender differences in performance. We suspect that female users, who typically have smaller fingers, might perform better. Second, to study fined-grained interactions related to usage patterns in the ZShift and Callout keyboards. For instance, do users initially touch some keys adjacent to their target? If so, do they succeed in correcting such mistake? Third, to incorporate some error correcting and/or prediction mechanisms. As previously discussed, these are used in modern soft keyboards, and for tiny screens it is plausible that they would be very useful.

Looking forward, we believe our work provides valuable research opportunities in text entry on very small screens. In general, manufacturers can benefit from this paper by selecting the most appropriate qwerty soft keyboard for their devices. Ultimately, this work provides designers, researchers, and practitioners with new understanding of qwerty soft keyboard design space and its scalability for tiny touchscreens.

ACKNOWLEDGMENTS

We thank our participants and the anonymous CHI reviewers. This work is part of the Valorization and I+D+i Resources program of VLC/CAMPUS and has been funded by the Spanish MECD as part of the International Excellence Campus program. This work has also been partially supported by the Spanish MINECO (TIN2014-37475 and TIN2010-20488) and the GVA VALi+d program (APOSTD/2013/013).

REFERENCES

1. Baudisch, P., and Chu, G. Back-of-device interaction allows creating very small touch devices. *Proc. CHI* (2009).
2. Bradley, J. V. Complete counterbalancing of immediate sequential effects in a Latin square design. *J. Amer. Statist. Ass. 53* (1958).
3. Chen, X., Grossman, T., and Fitzmaurice, G. Swipeboard: A text entry technique for ultra-small interfaces that supports novice to expert transitions. *Proc. UIST* (2014).
4. Dunlop, M., and Levine, J. Multidimensional pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking. *Proc. CHI* (2012).

5. Dunlop, M. D., and Crossan, A. Predictive text entry methods for mobile phones. *Personal Technologies 4*, 2–3 (2000).
6. Dunlop, M. D., and Masters, M. M. Pickup usability dominates: A brief history of mobile text entry research and adoption. *Int. J. Mobile Hum-Comput. Int. 1*, 1 (2009).
7. Felzer, T., and Nordmann, R. Alternative text entry using different input methods. *Proc. ASSETS* (2006).
8. Findlater, L., Wobbrock, J. O., and Wigdor, D. Typing on flat glass: examining ten-finger expert typing patterns on touch surfaces. *Proc. CHI* (2011).
9. Goldberg, D., and Richardson, C. Touch-typing with a stylus. *Proc. CHI* (1993).
10. Gunawardana, A., Paek, T., and Meek, C. Usability guided key-target resizing for soft keyboards. *Proc. IUI* (2010).
11. Harrison, C., and Hudson, S. E. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. *Proc. UIST* (2009).
12. Henze, N., Rukzio, E., and Boll, S. Observational and experimental investigation of typing behaviour using virtual keyboards for mobile devices. *Proc. CHI* (2012).
13. Himberg, J., Häkkilä, J., Kangas, P., and Mäntyjärvi, J. On-line personalization of a touch screen based keyboard. *Proc. IUI* (2003).
14. Ingmarsson, M., Dinka, D., and Zhai, S. TNT: a numeric keypad based text input method. *Proc. CHI* (2004).
15. Isokoski, P., and Raisamo, R. Device independent text input: a rationale and an example. *Proc. AVI* (2000).
16. James, C. L., and Reischel, K. M. Text input for mobile devices: comparing model prediction to actual performance. *Proc. CHI* (2001).
17. Kienzle, W., and Hinckley, K. Writing handwritten messages on a small touchscreen. *Proc. MobileHCI* (2013).
18. Kim, S., Sohn, M., Pak, J., and Lee, W. One-key keyboard: a very small QWERTY keyboard supporting text entry for wearable computing. *Proc. OZCHI* (2006).
19. Komninos, A., and Dunlop, M. Text input on a smart watch. *Pervasive Computing 13*, 4 (2014).
20. Kristensson, P. O., and Zhai, S. Relaxing stylus typing precision by geometric pattern matching. *Proc. IUI* (2005).
21. Leiva, L. A., and Sanchis-Trilles, G. Representatively memorable: Sampling the right phrase set to get the text entry experiment right. *Proc. CHI* (2014).
22. Li, F. C. Y., Guy, R. T., Yatani, K., and Truong, K. N. The 1line keyboard: a QWERTY layout in a single line. *Proc. UIST* (2011).
23. Lyons, K., Nguyen, D., Ashbrook, D., and White, S. Facet: a multi-segment wrist worn system. *Proc. UIST* (2012).
24. Lyons, K., Starner, T., Plaisted, D., Fusia, J., Lyons, A., Drew, A., and Looney, E. Twiddler typing: One-handed chording text entry for mobile phones. *Proc. CHI* (2004).
25. MacKenzie, I. S., and Soukoreff, R. W. Phrase sets for evaluating text entry techniques. *Proc. CHI EA* (2003).
26. MacKenzie, I. S., Soukoreff, R. W., and Helga, J. 1 thumb, 4 buttons, 20 words per minute: design and evaluation of H4-writer. *Proc. UIST* (2011).
27. MacKenzie, I. S., and Zhang, S. X. The immediate usability of graffiti. *Proc. GI* (1997).
28. MacKenzie, I. S., and Zhang, S. X. The design and evaluation of a high-performance soft keyboard. *Proc. CHI* (1999).
29. Oney, S., Harrison, C., Ogan, A., and Wiese, J. ZoomBoard: a diminutive QWERTY soft keyboard using iterative zooming for ultra-small devices. *Proc. CHI* (2013).
30. Oulasvirta, A., Reichel, A., Li, W., Zhang, Y., Bachynskyi, M., Vertanen, K., and Kristensson, P. O. Improving two-thumb text entry on touchscreen devices. *Proc. CHI* (2013).
31. Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G., and Want, R. TiltType: accelerometer-supported text entry for very small devices. *Proc. UIST* (2002).
32. Perlin, K. Quikwriting: continuous stylus-based text entry. *Proc. UIST* (1998).
33. Ren, X., and Moriya, S. Improving selection performance on pen-based systems: a study of pen-based interaction for selection tasks. *ACM TOCHI 7*, 3 (2000).
34. Roeber, H., Bacus, J., and Tomasi, C. Typing in thin air: the canesta projection keyboard – a new method of interaction with electronic devices. *Proc. CHI EA* (2003).
35. Roudaut, A., Huot, S., and Lecolinet, E. TapTap and MagStick: improving one-handed target acquisition on small touch-screens. *Proc. AVI* (2008).
36. Sazawal, V., Want, R., and Borriello, G. The unigesture approach one-handed text entry for small devices. *Proc. MobileHCI* (2002).
37. Soukoreff, R. W., and MacKenzie, I. S. Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. *Proc. CHI* (2003).
38. Soukoreff, R. W., and MacKenzie, I. S. Recent developments in text-entry error rate measurement. *Proc. CHI* (2004).
39. Vogel, D., and Baudisch, P. Shift: A technique for operating pen-based interfaces using touch. *Proc. CHI* (2007).
40. Wobbrock, J. O., and Myers, B. A. Analyzing the input stream for character-level errors in unconstrained text entry evaluations. *ACM TOCHI 13*, 4 (2006).
41. Wobbrock, J. O., Myers, B. A., and Kembel, J. A. EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. *Proc. UIST* (2003).
42. Xiao, R., Laput, G., and Harrison, C. Expanding the input expressivity of smartwatches with mechanical pan, twist, tilt and click. *Proc. CHI* (2014).
43. Yatani, K., Partridge, K., Bern, M., and Newman, M. Escape: a target selection technique using visually-cued gestures. *Proc. CHI* (2008).
44. Zhai, S., Hunter, M., and Smith, B. A. Performance optimization of virtual keyboards. *Hum-Comput. Interact.* (2002).
45. Zou, Y., Liu, Y., Liu, Y., and Wang, K. Overlapped handwriting input on mobile phones. *Proc. ICDAR* (2011).