Invisiboard: Maximizing Display and Input Space with a Full Screen Text Entry Method for Smartwatches

Aske Mottelson, Christoffer Larsen, Mikkel Lyderik, Paul Strohmeier & Jarrod Knibbe
Department of Computer Science, University of Copenhagen
Njalsgade 128, DK-2300 Copenhagen, Denmark
{amot, cwl, mikl, p.strohmeier, jarrod}@di.ku.dk

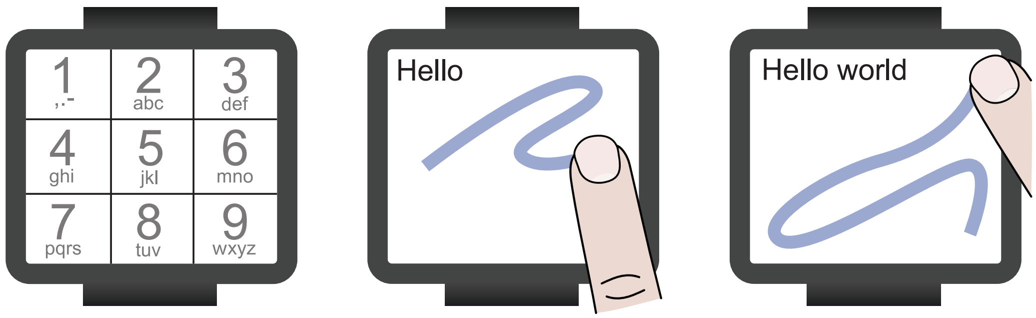


Figure 1: Invisiboard combines T9 text entry with swiping, to enable fast text entry on tiny screens while maximizing display space. Left: The screen is divided into nine areas, each containing 3-4 characters. Middle, Right: During use, the keyboard remains invisible. The user swipes across the desired keys and the software presents words based on a language model. If necessary, the user can scroll through available words, ordered by probability.

ABSTRACT

The small displays of smartwatches make text entry difficult and time consuming. While text entry rates can be increased, this continues to occur at the expense of available screen display space. Soft keyboards can easily use half the display space of tiny-screened devices. To combat this problem, we present Invisiboard: an invisible text entry method using the entire display for both text entry and display simultaneously. Invisiboard combines a numberpad-like layout with swipe gestures. This maximizes input target size, provides a familiar layout, and maximizes display space. Through this, Invisiboard achieves entry rates comparable or even faster than an existing research baseline. A user study with 12 participants writing 3264 words revealed an entry rate of 10.6 Words Per Minute (WPM) after 30 minutes, 7% faster than ZoomBoard. Furthermore, with nominal training, some participants demonstrated entry rates of over 30 WPM.

ACM Classification Keywords

H.5.2 User Interfaces: Input devices and strategies

Author Keywords

Text entry; Smartwatch; Wearables; Swipe

MobileHCI '16, September 06-09, 2016, Florence, Italy
©2016 ACM. ISBN 978-1-4503-4408-1/16/09…$15.00
DOI: http://dx.doi.org/10.1145/2935334.2935360

INTRODUCTION

Wearable technologies such as the Sony SmartWatch, the Apple Watch, or head-mounted displays, such as Google Glass, continue to receive consumer attention, but their use is heavily constrained by their small displays [8]. Text entry in particular is both slow and inaccurate using regular soft keyboard techniques on these devices.

Traditionally, text entry techniques for small displays employ QWERTY-like designs [2, 8, 14], thus allocating discrete, small input areas for every key. These input areas are difficult to target, making text entry slow and cumbersome. In order to address this, methods have been studied that require multiple taps for selecting a single letter [2, 4, 14] or memorization of individual gestures [11, 19]. Additionally, many approaches further constrain the already limited input area by dividing the display into separate display and input areas [2, 4, 8, 14].

Our design contribution, Invisiboard, presents a novel approach to overcome these restrictions. Invisiboard takes advantage of the full screen area for both display and text input simultaneously. A semi-transparent overlay guides novice users' input (see Figure 2a). As their expertise develops, the keyboard can become fully transparent (see Figure 2b). By using the whole display, Invisiboard provides large target sizes for text entry. The screen is divided into nine equally-sized invisible areas, thereby allocating increased space for each key compared to QWERTY-based keyboards. The alphabetical layout of the letters, inspired by T9 [3] is familiar, thus removing the requirement to learn new mappings of letters to unfamiliar gestures. The use of swipe gestures with a language model, inspired by the Swype keyboard [5], reduces the need for input accuracy across multiple taps.

We show that our proposed text entry method is easy to learn and fast to use, achieving word entry rates at least in line with a state-of-the-art input method, ZoomBoard [14] (see Figure 2c). Furthermore, the top entry rates measured with Invisiboard outperformed top rates of ZoomBoard by a factor two. Thus, Invisiboard at least maintains existing state-of-the-art text entry rates on wearable devices, while doubling the effective available display space.

RELATED WORK

Text entry for mobile devices has been an active research area for far longer than the recent wave of wearable technologies. Dial phones already mapped letters to numbers in the 1950's [17]. Early text entry methods used in HCI include number sequences for letter input on mobile phones [15].

As the screen sizes of mobile devices increased and physical interfaces disappeared, mobile text entry techniques were discarded in favor of soft QWERTY keyboards, based on well-known desktop computer interaction. Research then persisted this input style over to small screen devices, such as smartwatches [2, 4, 14].

QWERTY-based Text Entry

Accurate text selection with QWERTY-based keyboards on small devices is challenging, requiring precise targeting [14]. To make targeting easier, recent work introduced a two step selection process: first the user targets the general area of the keyboard, selecting the exact letter in a second step.

When typing with ZoomBoard by Oney et al. [14] users tap the keyboard to see a zoomed-in view of that area. This allows the user to then select the desired character with higher precision. SplitBoard by Hong et al. [4], further simplifies this process by presenting the user with two predefined zoomed-in areas: the keyboard layout is horizontally split in two sections such that it requires the user to flick to either half of the soft keyboard to locate a character. A study showed that SplitBoard outperformed the entry speed of ZoomBoard, although with a higher error rate [4].

A slightly more elaborate approach is Swipeboard by Chen et al. [2]. Their keyboard is divided into nine QWERTY-based regions. An initial swipe determines the selected region (tapping selects the centered region), and the user is then presented with the 3-4 characters. A second swipe determines the selected letter. In a comparison study, Swipeboard outperformed ZoomBoard in entry rate, with a similar error rate.

Other QWERTY-based soft keyboard layouts for tiny touch devices include Callout [8], ZShift [8], QLKP [4], SlideBoard [4], and VSQ [1]. Compared to ZoomBoard, Callout and ZShift obtained slower to comparable input speeds depending on display sizes, while QLKP, SlideBoard, and VSQ obtained higher input speeds than ZoomBoard.

A fundamental challenge for QWERTY-based input on small displays is the requirement for many discrete visual input controls; making targeting cumbersome, and sacrificing display space. Although different techniques have attempted to make targeting easier, with swiping [2, 5], zooming [14], and sliding [4], the visual representations have remained as a guide. Conversely, non-QWERTY based input techniques have been presented as a means of reducing input controls.

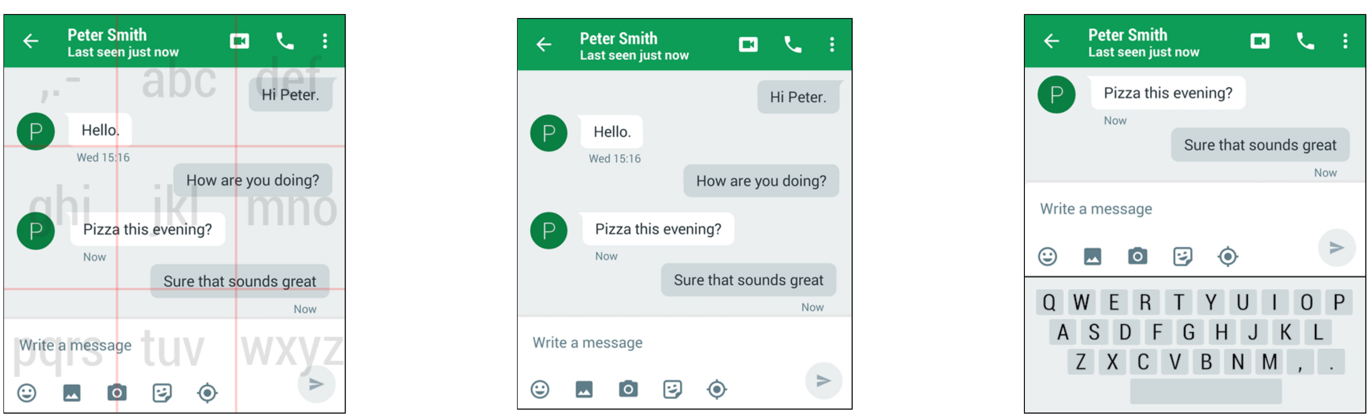
Non-QWERTY-based Text Entry

One of the first popularized text entry methods for mobile phones was T9 [3]. T9 used the letter to number mapping introduced by the dial phone together with a word recognition algorithm. T9 supported expert users: users who sent five or more text messages per week over a period of six months were able to achieve writing speeds of over 20 WPM [6].

Alternatives to QWERTY-based input methods also include H4 by MacKenzie et al. [11] and EdgeWrite by Wobbrock et al. [19]. These techniques are able to achieve very fast text input speeds, but require substantial memorization to master and are difficult for novice users due to unfamiliar input layouts.

Minimizing Text Entry Visuals

Some keyboard designs have begun to address the issue of visuals obscuring large parts of the limited input space, while maintaining well-known input styles. Both the 1Line Keyboard [9] and Minuum [13] reduce the impact of keyboard



(a) (b) (c)

Figure 2: Google Hangouts using (a) Invisiboard with the semi-transparent overlay, (b) Inivisiboard without visual cues, and (c) ZoomBoard.

visuals on the display space by proposing compact QWERTY-like keyboards fitted to approximately one line at the bottom of the display area. Although these keyboards begin to reduce the on-screen impact of the controls, no previous work has completely minimized the input controls while supporting fast, accurate, and easy-to-learn text entry.

Across all of the input techniques intended for tiny devices, the keyboard visuals place additional burden on the devices' already limited screen space. In this paper, we suggest that maximizing text entry rate and minimizing error rate should not be the only principle criteria of text entry techniques for tiny-screened devices. We argue that maximizing display space is equally important; making possible the entry of multi-line prose.

USER INTERFACE

The Invisiboard design is inspired by the T9 input technique [3]. Instead of dedicated buttons, Invisiboard divides the entire screen area into nine invisible regions such that the keyboard and display area are overlapping (see Figure 1 and Figure 2a-2b). These nine regions each represent a T9 component, containing sequential subsets of the alphabet. The second area represents the letters 'abc', the third 'def' and so forth. The first area contains a set of special characters.

If the user needs help remembering the location of a letter, a two-finger tap enables a semi-transparent overlay on top of the input area, with a visible division of the T9 components (see Figure 2a).

To input a word, the user swipes across each area in turn that contains a requisite letter. The user leaves behind a swipe 'path' representing groups of letters, which is then fitted to a language model. Whenever the user lifts their finger from the touch screen, the recognizer predicts the intended work and enters it, followed by a space. A direct consequence of using this approach is that ambiguous results for inputted sequences exist, for instance *if* and *he* are both constructed by the sequence '43'. If the intended word is not the initial predicted result, users can scroll through corresponding results to select their word. We tried two approaches for scrolling through ambiguities: scrolling on the right edge and using a *fat swipe* (with a finger pad, instead of tip). A pilot study showed preference towards using the latter. Deleting words is done with a *swipe-left* gesture at the bottom edge of the screen.

IMPLEMENTATION

Word prediction model

Our dictionary contains 50,000 words and their natural frequencies. The numerical representation of every word was computed, ignoring repeated characters. For example, *hello* becomes '4356'. An optimal line segment for every word was also computed. This is the line intersecting the centers of every area contained in the numerical representation of the word. The words along with their optimal line segments are stored in a hash map structure, using the numerical first and last values as keys, such that words that start and end in the same region are stored at the same key. This way we optimize our prediction quality and reduce computational complexity - only considering potential matches that start and end with the same 'keys.' This idea is based on the observation that users' inputs are commonly correctly initiated and ended (Figure 4a), but traverse several unintended areas during the interaction.

To minimize the number of scrolls required by users, we first order applicable words by their similarity scores using the recognizer described below. The 12 best fits are then ordered by their natural frequencies to promote common words. This is a somewhat rudimentary approach, and a future iteration of the recognizer could employ a more principled approach. Kristensson and Zhai [7] showed how to integrate probabilities from multiple channels in gesture based text-entry. By using Bayes' rule as [7] we could obtain confidence scores for each word matching a gesture, integrating both the spatial similarities and word frequencies.

Gesture recognizer

We sought a unistroke recognizer that was direction-, rotation-, and scale-variant. Our developed gesture recognizer works by comparing inputted line segments to the pre-computed optimal templates. Line segments of inputs and templates are resampled into the same number of equidistant points. The similarity between an input and a template is then calculated as the sum of the Euclidean distances between all pairs of points, inspired by Vatavu et al. [18]:

$$\sum_{i=1}^{n} ||I_i - O_i|| = \sum_{i=1}^{n} \sqrt{(x_{inp} - x_{opt})^2 + (y_{inp} - y_{opt})^2}$$

where I is the user inputted line segment transformed to some (x_{inp}, y_{inp}) equidistant points. O is the optimal path of (x_{opt}, y_{opt}) points constructed from the centroids of the relevant T9 areas for a candidate word.

EVALUATION

We carried out an evaluation in order to test the viability of our text entry design and to get metrics of novice users' performances using Invisiboard compared to ZoomBoard. ZoomBoard was chosen as the baseline since it offers users a proven easy-to-learn entry method for tiny devices. Additionally, ZoomBoard is frequently employed as a baseline in the related literature (e.g., [2, 4, 8]).

Participants

We recruited 12 participants (7 females), ages 22-32 (M = 24), all right-handed. Participants were compensated an equivalent of US $15. None of the participants had prior experience with smartwatches, one had experience with entering text by swiping, and 10 had prior experience with T9. None had English as their first language, but all participants were confident English speakers.

Apparatus

We carried out the experiment on a Nexus 5 Android smartphone, using two similar experimental applications employing Invisiboard and ZoomBoard, respectively. We designed the applications such that an area indicative of those seen on smartwatches (42 $\times$ 35.9 mm) covered the center of the smartphone screen. As our participants were novice users of both Invisiboard and the swiping technique, they used the semi-transparent overlay during the evaluation (see Figure 2a).

Design of Experiment

The evaluation was carried out as a within-subjects design; participants used both input techniques with balanced order. The experimental software presented participants with sentences from the MacKenzie and Soukoreff phrase set [10]. Users were then instructed to input a presented sentence and hit the next button upon completion. To imitate smartwatch usage, participants were instructed to hold the device in their non-dominant hand, and only interact with their dominant index finger.

Participants completed a session for both input techniques consisting of 32 trials, each trial consisting of 1 sentence. To avoid fatigue, the trials were grouped into 4 blocks of 8 trials. Thus, the participants were able to rest briefly once between the blocks (after 8 sentences) and also once between the sessions (after 32 sentences). The experiment lasted approximately 80 minutes per participant. In summary the experiment comprised: 12 participants $\times$ 2 input techniques $\times$ 4 blocks $\times$ 8 sentences $\times$ 4.25 words per sentence $=$ 3264 transcribed words.

Experimental Measures

The independent variable was *input technique*. We collected the raw touch data to calculate dependent variables which are described below.

Input Speed

The time spent inputting a sentence was calculated as the duration between the first and last interaction. As such, time spent reading sentences did not influence WPM. WPM was calculated using the following formula [16]:

$$\text{WPM} = \frac{|T|}{S} \times 60 \times \frac{1}{5}$$

$|T|$ is the length of the transcribed string, S is time in seconds. WPM does not account for mistakes in the transcribed sentence.

Error Rate

As noted by Markussen et al. [12], errors made with gesture-based keyboards (where the user inputs words, rather than characters) will result in wrong words. Errors made with traditional keyboards, such as QWERTY, will result in individual character errors. Therefore, as [12], the error rate was calculated using Minimum Word Distance (MWD), instead of Minimum String Distance (MSD):

$$\text{MWD error rate} = \frac{MWD(P,T)}{\overline{S_P}} \times 100\%$$

$\overline{S_P}$ is the mean length of the optimal alignments, calculated per-word. P and T are the sets of words in the presented and transcribed sentences. This error metric will cause higher error rates than MSD, because whole words are classified as wrong if a user produces a single letter mistake in a word.

RESULTS

We conducted a two-sample t-test assuming unequal variance for entry rate and error rate. We were not able to find a significant effect for entry rate or error rate.

The difference in input speed between the first and second half was significant for both input types. This tells us that the the learning effect was significant for both Invisiboard ($F_{1,382} = 16.4, p < .001$), and ZoomBoard ($F_{1,382} = 20.7, p < .001$). We therefore report results for the first and second half of the trials separately. Descriptive statistics can be found in Table 1.

Figure 3 demonstrates the input rate variance for both ZoomBoard and Invisiboard. Alongside supporting the fastest entry speeds, Invisiboard also recorded the slowest speeds, suggesting a steeper learning curve for some users.

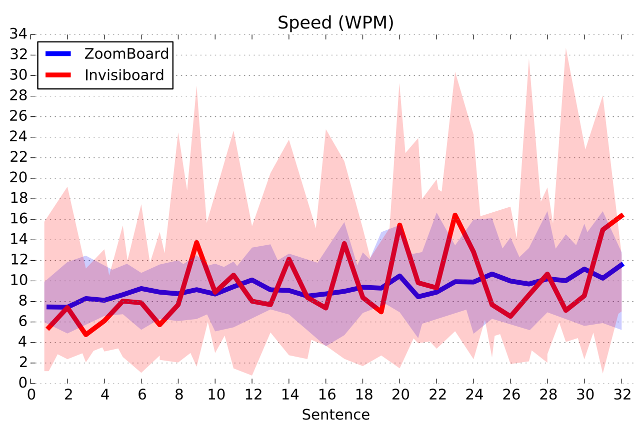


Figure 3: Figure showing average (line), maximum and minimum (solid colors) entry speeds (WPM) for both input techniques. Invisiboard (red) accounts for both the highest and lowest entry rates in the study.

Inspired by Kristensson and Zhai [7], in addition to reporting the formal evaluation statistics, we also report the maximum observed text entry speeds. This provides some indication of achievable input rates with further practice. Table 2 lists

	Entry rate (WPM)			**Error rate (MWD %)**		
	First half	Second half	Total	First half	Second half	Total
ZooomBoard	8.7 ± 1.8	9.9 ± 2.2	9.3 ± 2.0	2.2 ± 4.9	2.1 ± 4.1	2.1 ± 4.5
Invisiboard	8.2 ± 4.1	10.6 ± 4.9	9.5 ± 4.5	2.4 ± 5.7	3.9 ± 6.2	3.2 ± 6.0

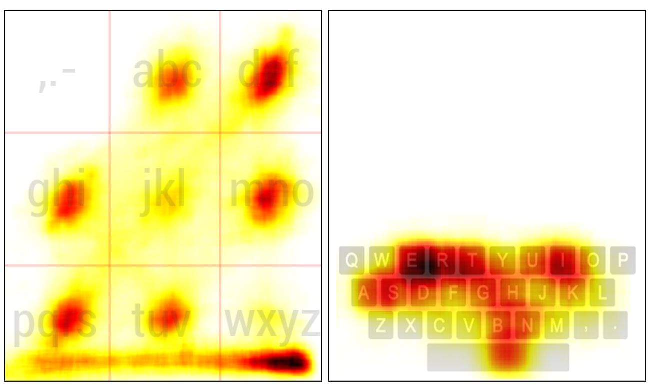
Table 1: Performance results from evaluation. Results are formatted as mean $\pm$ SD.

instances of the three fastest inputted sentences for both input techniques; the fastest inputted sentence with Invisiboard showed an entry rate of 33 WPM, while the fastest input speed registered with ZoomBoard was 16 WPM.

Phrase	Method	WPM
one heck of a question	Invisiboard	32.7
can I skate with sister today		31.7
great disturbance in the force		30.4
neither a borrower nor a lender be	ZoomBoard	16.1
beware the ides of March		16.0
he is just like everyone else		15.7

Table 2: Instances of the top three fastest inputted sentences for both input techniques. These findings show that experts of Invisiboard can input much faster compared to using existing tiny device input techniques.

We generated heat maps for both input techniques from the raw touch data, using the (x, y) positions of any user activity (see Figure 4a-4b). The heat maps thus show the spatial distribution of the aggregated touch activity of all users, including touches meant for corrections, zooming, selecting, and scrolling through word suggestions.



(a) **(b)**

Figure 4: The distribution of users' touches on top of (a) Invisiboard and (b) ZoomBoard.

DISCUSSION

Figure 4a-4b illustrate the additional complexity and targeting accuracy associated with the use of traditional QWERTY-based input approaches; looking at these heat maps we see that the QWERTY-based keyboard condenses interactions to a small part of the display. To improve ease of targeting, Invisiboard takes advantage of a much larger area of the display. In addition to making targeting easier, Invisiboard increased the available display space significantly.

The within participant variability in entry rate for Invisiboard was higher than for ZoomBoard (SD of 4.5 and 2.0 WPM respectively). In this way, Invisiboard challenged some, but leveraged high performance in others (see Figure 3). This difference was even more dominant in the second half of the trials (see Table 1).

When looking at the slowest entries with Invisiboard it becomes apparent that an extensive amount of time was spent on text entry correction and selection. If we remove the time spent on correcting entries (i.e., backspaces for ZoomBoard, and bottom swipes and scrolling for Invisiboard) the average input speed becomes 26.8 WPM (SD = 6.4) for Invisiboard and 11.2 WPM (SD = 1.7) for ZoomBoard. This shows that the potential for optimizing entry speeds with Invisiboard is vast. If a more advanced language model was employed, such as that of [7], entry speeds could potentially increase immensely. Additionally, a more seamless scrolling mechanism could improve performance, for instance by utilizing the crown available on many consumer smartwatches.

The variation in entry speed is further exacerbated when entering sentences containing uncommon words with Invisiboard (see, for example, the phrases in Table 2), as users have to scroll through a longer series of ambiguous suggestions. In real a life scenario, we would expect users to be writing simple, conversational messages on a smartwatch, rather than complex texts. For the 15% most common words (such as *you*, *that*, and *have*) Invisiboard outperformed ZoomBoard with 20%. So in practice, we believe the advantage of using Invisiboard instead of a QWERTY-based solution would be greater than what our user study demonstrates on average.

Looking at the learning effects witnessed between the two halves of the study with Invisiboard, it appears that an opportunity for further learning beyond the in-study usage exists. James and Reischel [6] showed that T9 leverages expert usage more than other entry methods, and in a similar manner, we believe that with continued use of Invisiboard its advantages will become more prominent. Not only will text entry rates increase, but once the user is familiar with Invisiboard, the visual guides are also no longer needed, and the interface can become truly invisible. Table 2 gives an estimation of the entry speed increase the transition from novice to expert user provides; the fastest inputted sentences using Invisiboard were twice the speed of ZoomBoard.

Other studies on text entry techniques for wearables also compared performances to ZoomBoard ([2, 4, 8]). In comparison to the input speed of ZoomBoard, Swipeboard [2] registered a 15% improvement, SplitBoard [4] a 34% improvement, and Callout and ZShift [8] reported similar to lower speeds. Since the performances reported in text-entry studies are highly dependent on the study design, the pre-task practice, the recruited participants, and the phrase set employed, a direct comparison between results found in the literature is problematic. For example, SplitBoard [4] and SwipeBoard [2] used a between subjects design, while Callout and ZShift [8] were compared to ZoomBoard in a within-subjects design. Additionally different phrase sets were employed, thus making the input difficulty across the studies fundamentally different; for example, Swipeboard attempted to simulate learning effects by reducing their character selection to 5 letters ('E', 'T', 'A', 'N', 'S').

Although other research reports a greater speed increase over ZoomBoard than we report here, these keyboards employ visually rich, QWERTY-like designs, sacrificing valuable screen space (see comparison of visible screen space in Table 3).

Through our work, we demonstrate the potential of Invisiboard as a fast input technique, whilst maximizing screen display space. For text entry on small screen devices, we argue that the available display space should be considered an important usability criterion, alongside the input speed and error rates.

Keyboard	Available Display Space
Invisiboard	100%
Minuum [13]	70%
ZoomBoard [14]	50%
SwipeBoard [2]	50%
ZShift, CallOut [8]	50%
SplitBoard, QLKP, SlideBoard [4]	25%
VSQ [1]	25%

Table 3: The available display space among keyboards for tiny displays. These numbers are estimated using image processing on the figures provided in the papers or from relevant sources. Screen space spent on word suggestions are not taken into account, as these could be presented directly in the text field, as with Invisiboard.

As Invisiboard offers an invisible interface, there needs to be clear indication to the user that the keyboard is currently enabled (as to not interfere with touch navigation or other GUI controls). We did not implement this feature for our study, but one can easily envisage a screen-border indicating that the keyboard is active, combined with a quick enable-/disable-keyboard clutching mechanism.

FUTURE WORK

The Invisiboard prototype does not currently support the input of words not contained in the dictionary, such as proper nouns. Other swipe gesture keyboards also feature regular tap entry (e.g., [5]), useful for inputs that are repeatedly misinterpreted. The T9 technique defaulted to a multi-tap technique to input unknown words. Future iterations of Invisiboard should consider such fallback techniques. The current implementation only offer a very limited set of special characters, and no numbers. In a future setup, we could implement a mode to switch between normal and special characters.

Improvement is also possible in the word recognition as roughly three quarters of the interactions and a third of the time using Invisiboard was spent on corrections of previous input. Input speeds can be dramatically improved with an optimized prediction engine; for instance using Bayes theorem [7], more templates, context analysis, and user dependent recognition.

A future study should investigate the effects of expertise to verify the assumptions stated of performance improvements during the transition from novice to expert. This study should also report on the effects of removing the semi-transparent visual cues.

CONCLUSION

With the continued interest in smartwatches and other miniaturized touch devices, efficient and effective text entry continues to be an important area of focus. Most previous input techniques in this domain considered alterations of QWERTY which require a large number of graphical control elements. Recent advances have increased the input speed of QWERTY keyboards on tiny-displays, but continue to sacrifice precious display space. In this paper we present a design that supports efficient text entry alongside full-screen display space. Invisiboard provides large target sizes for selection and provides a familiar layout. A study demonstrated that even with minimal training our interaction technique performs at least as well as an existing baseline technique, but shows potential to perform even faster with continued use.

While much can still be done to improve speed and ease of input in this domain, we have shown that non-QWERTY input techniques are well suited for small touch interfaces. We hope our exploration will pave the way for further investigations in text entry on small touch devices, taking advantage of alternative keyboard designs and interaction styles that continue to maximize the available display space.

ACKNOWLEDGEMENTS

This work was supported by the European Research Council, grant no 648785.

REFERENCES

1. Jae-Min Cha, Eunjung Choi, and Jihyoun Lim. 2015. Virtual Sliding QWERTY: A new text entry method for smartwatches using Tap-N-Drag. *Applied Ergonomics* 51 (2015), 263 – 272. DOI: http://dx.doi.org/10.1016/j.apergo.2015.05.008

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

3. D.L. Grover, M.T. King, and C.A. Kushler. 1998. Reduced keyboard disambiguating computer. (Oct. 6 1998). http://www.google.com/patents/US5818437 US Patent 5,818,437.

4. Jonggi Hong, Seongkook Heo, Poika Isokoski, and Geehyuk Lee. 2015. SplitBoard: A Simple Split Soft Keyboard for Wristwatch-sized Touch Screens. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 1233–1236. DOI: http://dx.doi.org/10.1145/2702123.2702273

5. Swype Inc. 2016. *"The Swype Virtual Keyboard"*. http://www.swype.com/. (2016).

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

7. 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
8. Luis A. Leiva, Alireza Sahami, Alejandro Catala, Niels Henze, and Albrecht Schmidt. 2015. Text Entry on Tiny QWERTY Soft Keyboards. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 669–678. DOI: http://dx.doi.org/10.1145/2702123.2702388
9. Frank Chun Yat Li, Richard T. Guy, Koji Yatani, and Khai N. Truong. 2011. The 1Line Keyboard: A QWERTY Layout in a Single Line. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 461–470. DOI: http://dx.doi.org/10.1145/2047196.2047257
10. 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
11. I. Scott MacKenzie, R. William Soukoreff, and Joanna Helga. 2011. 1 Thumb, 4 Buttons, 20 Words Per Minute: Design and Evaluation of H4-writer. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 471–480. DOI: http://dx.doi.org/10.1145/2047196.2047258
12. 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
13. Minuum. 2016. *"Minuum on Smartwatches"*. http://minuum.com/minuum-on-smartwatch/. (2016).
14. Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. 2013. ZoomBoard: A Diminutive Qwerty Soft Keyboard Using Iterative Zooming for Ultra-small Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 2799–2802. DOI: http://dx.doi.org/10.1145/2470654.2481387
15. Chris Schmandt. 1993. Phoneshell: The Telephone As Computer Terminal. In *Proceedings of the First ACM International Conference on Multimedia (MULTIMEDIA '93)*. ACM, New York, NY, USA, 373–382. DOI: http://dx.doi.org/10.1145/166266.168430
16. R. William Soukoreff and I. Scott MacKenzie. 2003. Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 113–120. DOI: http://dx.doi.org/10.1145/642611.642632
17. Bell Telephone Systems. 1951. *The Telephone and How We Use It*. Wayne University, Detroit.
18. Radu-Daniel Vatavu, Lisa Anthony, and Jacob O. Wobbrock. 2012. Gestures As Point Clouds: A $P Recognizer for User Interface Prototypes. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction (ICMI '12)*. ACM, New York, NY, USA, 273–280. DOI: http://dx.doi.org/10.1145/2388676.2388732
19. Jacob O. Wobbrock, Brad A. Myers, and John A. Kembel. 2003. EdgeWrite: A Stylus-based Text Entry Method Designed for High Accuracy and Stability of Motion. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST '03)*. ACM, New York, NY, USA, 61–70. DOI: http://dx.doi.org/10.1145/964696.964703