ZoomBoard: A Diminutive QWERTY Soft Keyboard Using Iterative Zooming for Ultra-Small Devices

Stephen Oney, Chris Harrison, Amy Ogan, Jason Wiese
Carnegie Mellon University & Friendship House
Pittsburgh, PA 15213 USA
{ soney, chris.harrison, aeo, wiese }@cs.cmu.edu

ABSTRACT
The proliferation of touchscreen devices has made soft keyboards a routine part of life. However, ultra-small computing platforms like the Sony SmartWatch and Apple iPod Nano lack a means of text entry. This limits their potential, despite the fact they are capable computers. In this work, we present a soft keyboard interaction technique called ZoomBoard that enables text entry on ultra-small devices. Our approach uses iterative zooming to enlarge otherwise impossibly tiny keys to comfortable size. We based our design on a QWERTY layout, so that it is immediately familiar to users and leverages existing skill. As the ultimate test, we ran a text entry experiment on a keyboard measuring just 16 x 6mm – smaller than a US penny. After eight practice trials, users achieved an average of 9.3 words per minute, with accuracy comparable to a full-sized physical keyboard. This compares favorably to existing mobile text input methods.

ACM Classification: H.5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces; Input devices and strategies.

General terms: Design, Human Factors.

Keywords: Text entry, mobile input, interaction technique, handheld device, fat finger, zooming user interfaces.

INTRODUCTION

With the proliferation of handheld touchscreen devices, soft keyboards have become an important text entry mechanism – a daily part of life for many people. This has prompted considerable research in the field of mobile text entry. There are two significant challenges any successful soft keyboard technique must overcome. Foremost, it has to be comfortable and accurate on the small confines of a mobile device. Secondly, the technique must be palatable to users, and ideally already familiar. Users are highly resistant to learning new methods, particularly new keyboard layouts [30].

We present our work on ZoomBoard, a novel text entry technique for inch-scale [25] and smaller devices, such as a wristwatch computers or digital jewelry [20]. These devices are computationally capable, but suffer from a paucity of input area, which generally precludes the use of conventional soft keyboards (and in turn limits the range of possible applications). At present, text entry methods for such small devices are absent (e.g., iPod Nano, Sony SmartWatch), unwieldy (e.g., calculator watch), or exotic (see e.g., [16,17,24]), which necessitates training. When such devices are worn on the body, social acceptability and comfort is directly proportional to device size [14], further prompting our research into ultra-small device text entry.

Our design objective was three fold: 1) To devise a method to enable text entry on *ultra-small* devices, while 2) being immediately *familiar* to users, and that 3) leveraged the thousands of hours of training users have with keyboards, such that *minimal training* is required. The resulting system uses a standard QWERTY layout, which iteratively zooms in response to user presses (Figure 1). As we will discuss, screens as small as 1” square are usable, achieving 9.3 words per minute (WPM). Qualitative results suggest users immediately understood the interaction and were able to start typing with minimal training.

RELATED SYSTEMS

Four areas of research inspired our work. One domain is high precision pointing on touchscreens (e.g., rubbing-pointing [22], ZoomPointing [1], multiscale pointing [10]). Also inspirational was seminal work on zooming user interface (ZUI) interaction paradigms, such as PAD [3] (see [6] for survey). Third, keyboard methods featuring iterative reduction of the keyset served as comparative design examples (e.g., LURD-writer [7], Dasher [24], TNT [13]). Finally, methods that employ zooming as an assistive technique (e.g., Zoom Screen [2], Lean and Zoom [11]).

The most direct approach to support text entry on mobile devices has been to find clever ways to shrink QWERTY keyboards, such that they can be accommodated on mobile

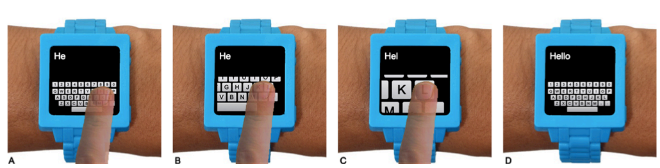

Figure 1: ZoomBoard on a watch-sized device. The keyboard is fully zoomed out by default (A). When users press a key, the keyboard iteratively zooms in (B & C), until the keys are a size that is comfortable and accurate (C). After the desired character is entered, the keyboard resets (D). Users may also swipe to the left to delete, to the right for a space, and up to switch to a symbols keyboard.

CHI 2013, April 27–May 2, 2013, Paris, France.

Figure 2: ZoomBoard could be used on a variety of device sizes, for example a smartphone (top left), watch (top right; simulated on a iPad), or coin-sized (bottom left). For size comparison, a typical physical number keypad (bottom right).

device enclosures and touchscreens [14]. However, fitting 27+ keys onto a small device is neither straightforward, nor particularly accurate. It is now standard for touchscreen keyboards to feature real-time spelling correction, simply because it is assumed that users are unable to accurately hit such small keys. However, this approach carries the significant benefit of being instantly familiar to users. Indeed, the latter property has been a driving factor in the consumer space, and most text produced today on smart devices is by miniature QWERTY keyboards. Further, given the number of keys required and our "fat fingers", it seems unlikely a full keyboard layout can be shrunk much further.

A second approach is to leave the QWERTY keyboard design behind. Despite user resistance [30], this approach is popular because performance gains can be significant. This has yielded a variety of alternative keyboard layouts, including ATOMIK [29], Metropolis [30], HandyKey Twiddler [16] and OPTI [19], among many others. Escaping the keyboard metaphor entirely, TiltText allowed users to enter characters through a combination of buttons and tilting a handheld device in one of four directions [26]. Dasher [24] lets users steer continuously through an alphabet with a cursor or touch; letter size is weighted by likelihood. EdgeWrite [27], Unistrokes [9], and Graffiti [5] use stylized characters written with a stylus; strokes are based on conventional written characters, which reduce the learning burden. Ni and Baudisch [21] considered text entry on "disappearing devices", and put forward a method of writing characters on top of a small optical motion sensor. The EdgeWrite gesture set was subsequently extended to four buttons [28], a potentially very small input platform that other researchers have also explored for text input [7,17].

Researchers have also attempted to strike a balance, retaining or tweaking the QWERTY layout, so as to leverage users existing knowledge, but introducing subtle usability improvements. For example, projects have looked at "quasi-QWERTY" [4] and "QWERTY-like" [12] layouts, allowing keys to be relocated a limited distance from their traditional positions. Efforts have also looked at compressing the QWERTY keyboard down into a single row of keys [15], while retaining the overall layout, so as to take up less screen real estate (though not less width).

ZOOMBOARD

ZoomBoard provides a full QWERTY keyset in a conventional configuration (Figure 1 and 2). To type, users press on a desired key. Because the keys are so small at the scales we are designing for, it would be highly inaccurate to immediately select a key. For reference, a typical smartphone soft keyboard is approximately 2" wide and even with a language model, key errors are made. Instead of immediate selection, our keyboard zooms in (Figure 1B). A smooth zooming transition is used to preserve perceptual constancy [23]. Next, with larger targets, the user can refine their finger position if needed, and once again press their desired key. If necessary, more levels of zoom can be employed (e.g., three levels, as shown in Figure 1). Once keys have reached a size that enables accurate section, zooming stops and the key is typed upon pressing.

Additionally, capital letters can be typed by pressing and momentarily holding a key. Access to non-alphanumeric keys is provided via three swipe gestures (see Video Figure). A swipe to the left deletes the last character. An upward swipe brings up a secondary keyboard with symbols. A swipe to the right types a space. We also included a functional space bar on the ZoomBoard layout primarily to act as an additional visual cue, though this could be removed to save space.

We experimented with three zooming strategies, depicted in Figure 3. Most successful in piloting was a linear combination of a zoom-centering approach with a "least finger movement strategy", where upon pressing the screen, the keyboard zooms, leaving the target directly under the finger. Anecdotally, most mapping applications on touch-screen devices use the latter paradigm for zooming.

ZoomBoard can be scaled to variety of screen sizes and shapes. For example, ZoomBoard could prove useful on the sixth generation iPod Nano - one of the smallest commercially available touchscreen devices, featuring a 1.54" diagonal screen (roughly 1" square). The just released seventh generation iPod Nano features a much larger 2.5" screen, but still lacks a text entry method.

ZoomBoard may also have implications for users with impaired vision and/or reduced motor skill. The zooming interaction could be applied to larger keyboards, like those found on smartphones and tablets, to further increase their size. We created a smartphone-sized version (Figure 2, top left) with one level of zoom. Although doubling the keystrokes per character (KSPC), buttons surface area increases 400%, providing both a larger item to see and target to hit.

Of note, ZoomBoard has an *innate* KSPC of 2.0 - that is, all keys can be typed with 2 presses. Spaces can be typed with one swipe, bringing the KSPC-min down to 1.84 on typical phrase sets. This compares favorably to other small-device input methods, for example, 2.1, 3.3, and 4.3 for H4-Writer

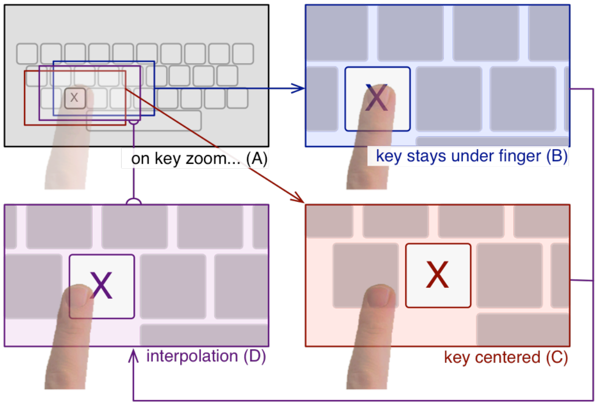

Figure 3: Three applicable zooming approaches. When the user zooms in on a key (A), the area of the keyboard pressed could stay under the finger (B) or move to the center (C). We use a linear combination of both (D).

[17], LURD Writer [7], and EdgeWrite [27,28] respectively (see [17] for more discussion).

As a proof-of-concept, we built a prototype of ZoomBoard in JavaScript for Webkit-based browsers. Our implementation uses native touch events on iOS devices (iPad and iPhone) to detect swipes and button presses (though our design could be ported to other platforms).

EVALUATION

To evaluate the feasibility of ZoomBoard, we used a standard text entry experimental design, which allows us to compare our approach to others through established metrics. Six participants (4 female and 2 male, μ=23.5 years old) were recruited from our university, and were given a small gratuity for their time. Each participant completed four sessions over two days, each of which involved multiple trials, described below.

In the first session, participants completed four trials using three different keyboard setups. First, they were administered a three-minute typing proficiency test on a standard physical keyboard. Next, participants were presented with a small non-zooming keyboard, displayed on an iPad 3. We instructed participants to type each phrase that appeared on the screen using the touchscreen keyboard. This trial lasted three minutes. Finally, participants were presented with the ZoomBoard. We instructed participants that their first tap on the ZoomBoard would zoom in to the corresponding area of the keyboard, and that the second tap would select the letter that would be typed. Participants completed two three-minute trials with the ZoomBoard.

Each participant completed three more sessions: one on the same day and two on the following day, with sessions occurring no less than two hours after the previous session. In each of the subsequent sessions, participants completed two three-minute trials with the ZoomBoard. In all trials, participants were instructed to correct their mistakes as they went. However, if they did not detect that a mistake was made until several characters later, they then should ignore the mistake and continue. At the end of Session Four, participants completed a short qualitative survey.

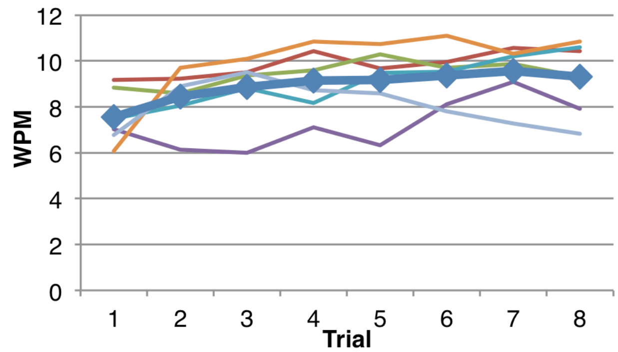

Figure 4: Participants' performance in wpm over the 8 trials using ZoomBoard. Mean performance shown in heavy blue line.

Both the small non-zooming keyboard and ZoomBoard measured 16.5 x 6.1mm; keys were 1.5 x 1.5mm (Figure 2, bottom left). ZoomBoard employed one level of zoom, enlarging keys to 4.4 x 4.4mm. Test phrases were drawn randomly from the 500-phrase corpus described by MacKenzie and Soukoreff [18].

RESULTS AND DISCUSSION

Participants achieved, on average, 7.6 wpm on their first use of ZoomBoard, and 9.3 wpm by the final trial (Figure 4). Furthermore, participants were able to achieve this entry rate with minimal impact on accuracy (mean KSPC of 2.15, SD=0.35). Transcribed phrases (mean length of 28.6 characters) on average contained 0.2 incorrect characters (Levenshtein distance), or one error per 1,430 characters.

ZoomBoard occupies a unique application space to which there is no immediate experimental comparison. Notable compact or mobile text input systems include Mulitap (11.0 wpm), Graffiti (11.4), TiltText (13.6), Unistroke (15.8), 4-key EdgeWrite (16.9), TNT (17.7), and H4-Writer (20.4) [5,13,17,26,28]. However, these systems use static, physical buttons or styli, and are several times larger than ZoomBoard (e.g., the latter three systems use keyboards, of which a single key is larger than our entire system). The fact that ZoomBoard approaches these entry speeds, along with competitive accuracy, suggests the approach offers a viable solution for text-entry on ultra-small devices.

Comparing against our baseline, participants using the non-zooming keyboard achieved 4.5 wpm. Further, on average, participants were much less accurate (mean KSPC of 2.86, SD=0.98). Some users, apparently frustrated at the lack of progress, neglected to correct errors at all, driving up the mean-string distance (MSD) error to 4.07 characters (vs. 0.2 with ZoomBoard). Put plainly, on average, 14% of the phrase was incorrect. These results suggest that a keyboard of this size, without zooming, is simply impractical.

Indeed, if we examine the difference between the trials in which participants used the non-zooming keyboard (*NoZoom*), and their first and last trials using ZoomBoard (*FirstZoom*, *LastZoom*), there is a significant performance difference in wpm ($F(2,15) = 20.15$, $p<.001$). A post-hoc Tukey's test showed that *NoZoom* was significantly worse than both *FirstZoom* ($p=0.003$) and *LastZoom* ($p<0.001$).

We also ran an ANOVA to investigate the differences between KSPC-extra in these *Physical*, *NoZoom*, *FirstZoom*, and *LastZoom* conditions, with participant as a random factor. The overall model showed statistically significant differences ($F(3,20)=13.82$, $p<.001$). A post-hoc Tukey's test showed that *NoZoom* was significantly higher than the full keyboard, *FirstZoom*, and *LastZoom* (all $p<0.001$). There was no significant difference between the *physical* keyboard, *FirstZoom* or *LastZoom*, indicating that ZoomBoard is about as accurate as a full-sized physical keyboard.

Qualitative results from the survey at the end of the study support our quantitative data. Participants expressed that, while they would not want to use ZoomBoard on their current touchscreen device (4/6 disagree or strongly disagree), they think ZoomBoard would be useful on a smaller device (5/6 agree or strongly agree). For the statement "I was comfortably using ___ keyboard," participants were generally not comfortable using the *non-zooming keyboard* (4/6 disagree or strong disagree), but were more satisfied with the *zooming keyboard* (0/6 disagree or strong disagree).

CONCLUSIONS AND FUTURE WORK

Results from our evaluation paint a clear picture of the usability of ZoomBoard. Although a smartphone-sized soft keyboard is clearly preferable, ZoomBoard provides a viable means for ultra-small devices to support text entry. Even with a small screen, it would still be useful to e.g., write small text messages, search for nearby restaurants, and get directions to an address.

There are several obvious extensions of ZoomBoard that would further boost performance. Most immediate is to incorporate a language model, which is standard practice on modern soft keyboards. This could not only probabilistically weight key presses at the final level of zoom, but also adjust the centering point of the first zoom step.

Finally, we selected a set of ZoomBoard parameters based on piloting, but it is possible other settings would yield superior performance. ZoomBoard is also ripe for automatically adapting to a user's skill and motor performance [8]. As mentioned previously, zooming buttons could mitigate inaccurate motor performance (by virtue of becoming bigger). Similarly, motor performance decreases when a user is in motion (e.g., while jogging). This context could be automatically detected by a smartphone (e.g., using accelerometers), and could activate a ZoomBoard keyboard mode.

REFERENCES

1. Albinsson, P. and Zhai, S. High precision touchscreen interaction. In *Proc. CHI '03*. 105-112.
2. Bates, R. and Istance, H. Zooming interfaces!: enhancing the performance of eye controlled pointing devices. In *Proc. ASSETS '02*. 119-126.
3. Perlin K., and Fox, D. Pad: an alternative approach to the computer interface. In *Proc. SIGGRAPH '93*. 57-64.
4. Bi, X., Smith, B. A., and Zhai, S. Quasi-qwerty soft keyboard optimization. In *Proc. CHI '10*. 283-286.
5. Castellucci, S. and MacKenzie, S. Graffiti vs. unistrokes: an empirical comparison. In *Proc. CHI '08*. 305-308.
6. Cockburn, A., Karlson, A., and Bederson, B.B. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.* 41,1, (January 2009).
7. Felzer, T. and Nordmann, R., Alternative text entry using different input methods. In *Proc. ASSETS '06*. 10-17.
8. Gajos, K.Z., Wobbrock, J.O. and Weld, D.S. Automatically generating user interfaces adapted to users' motor and vision capabilities. In *Proc. UIST '07*. 231-240.
9. Goldberg, D. and Richardson, C. Touch-typing with a stylus. In *Proc. CHI '93*. 80-87.
10. Guiard, Y., M. Beaudouin-Lafon, and D. Mottet. Navigation as multiscale pointing: Extending fitts' model to very high precision tasks. In *Proc. CHI 1999*. 450-457.
11. Harrison, C. and Dey, A.K. Lean and zoom: proximity-aware user interface and content magnification. *Proc. CHI '08*. 507-510.
12. Hwang S. and Lee, G. Qwerty-like 3x4 keypad layouts for mobile phone. In *CHI EA '05*. 1479-1482.
13. Ingmarsson, M., Dinka, D., and Zhai, S. TNT: a numeric keypad based text input method. In *Proc. CHI '04*. 639-646.
14. Kim, S., Sohn, M., Pak, J., and Lee, W. One-key keyboard: a very small QWERTY keyboard supporting text entry for wearable computing. In *Proc. OZCHI '06*. 305-308.
15. Li, F., Guy, R.T., Yatani, K., and Truong, K. N. The 1line keyboard: a QWERTY layout in a single line. In *Proc. UIST '11*. 461-470.
16. Lyons, K., Starner, T., Plaisted, D., Fusia, J., Lyons, A., Drew, A., and Looney, E. W. Twiddler typing: one-handed chording text entry for mobile phones. In *Proc. CHI '04*. 671-678.
17. MacKenzie, I. S., Soukoreff, R. W., and Helga, J. 1 thumb, 4 buttons, 20 words per minute: design and evaluation of H4-writer. In *Proc. UIST '11*. 471-480.
18. MacKenzie, I.S. and Soukoreff, R.W. Phrase sets for evaluating text entry techniques. In *CHI EA '03*. 754–755.
19. MacKenzie, I.S., and Zhang, S.X. The design and evaluation of a high-performance soft keyboard. In *Proc. CHI '99*. 25-31.
20. Miner, C. S., Chan, D. M. and Campbell, C. Digital jewelry: wearable technology for everyday life. In *CHI EA '01*. 45-46.
21. Ni, T. and Baudisch, P. Disappearing mobile devices. In *Proc. UIST '09*. 101-110.
22. Olwal, A., Feiner, S., and Heyman, S. Rubbing and tapping for precise and rapid selection on touch-screen displays. In *Proc. CHI '08*. 295-304.
23. Robertson, G.G., Card, S.K., Mackinlay, J.D. Information Visualization Using 3D Interactive Animation. *CACM*, 36(4), 1993. 56-71.
24. Ward, D.J., Blackwell, A.F., and MacKay, D.J.C. Dasher – a data entry interface using continuous gestures and language models. In *Proc. UIST '00*. 129-137.
25. Weiser, M. The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun*. Rev. 3, 3 (July 1999), 3-11.
26. Wigdor, D. and Balakrishnan, R. TiltText: using tilt for text input to mobile phones. In *Proc. UIST '03*. 81-90.
27. 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. In *Proc. UIST '03*. 61-70.
28. Wobbrock, J.O., Myers, B.A., and Rothrock, B. Few-key text entry revisited: mnemonic gestures on four keys. In *Proc. CHI '06*. 489-492.
29. Zhai, S., Hunter, M. and Smith, B.A. Performance optimization of virtual keyboards. *Hum.-Comput. Interact*. 17, 2,3 (2002). 89-129.
30. Zhai, S., Hunter, M., and Smith, B.A. The metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design. In *Proc. UIST '00*. 119-128.