# The 1Line Keyboard: A QWERTY Layout in a Single Line

*Frank Chun Yat Li, Richard T. Guy, Koji Yatani, and Khai N. Truong*
Department of Computer Science
University of Toronto
Toronto, ON M5S 3G4, Canada
{frankli, guy, koji, khai}@dgp.toronto.edu

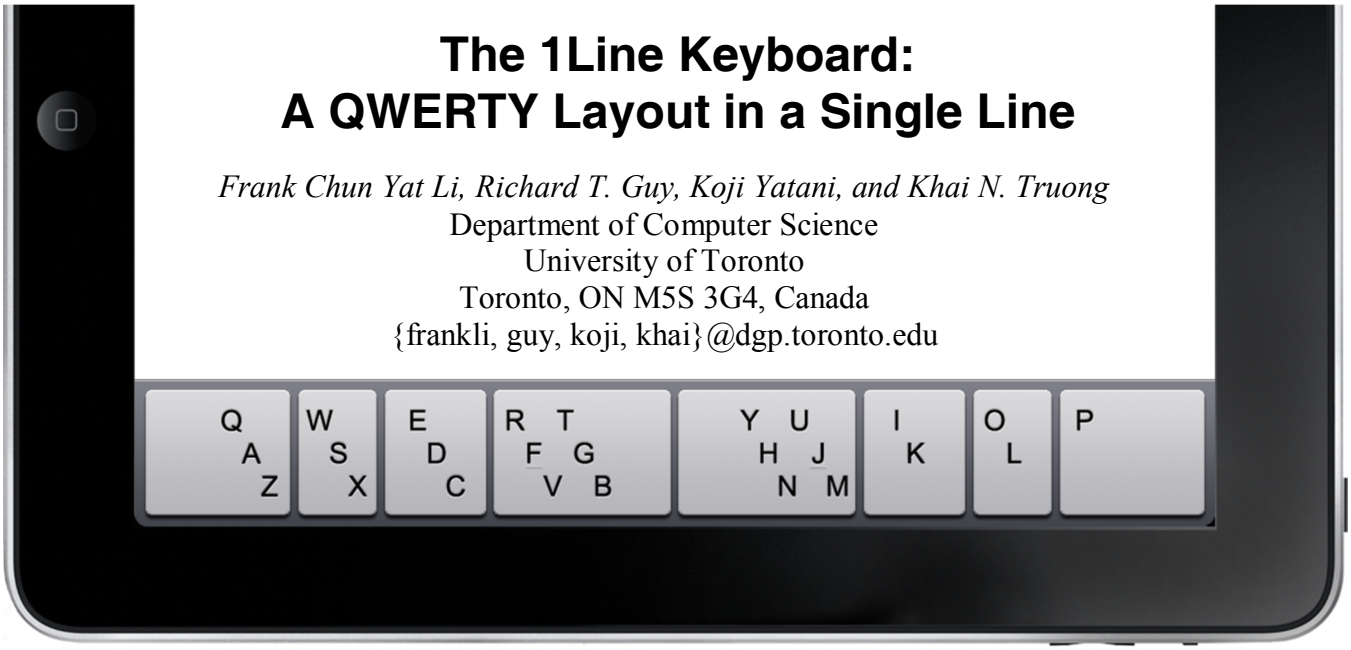


Figure 1: The 1Line Keyboard. It consists of only eight character keys, flick gestures, and a novel approach for integrating the spacebar into the bezel. It is 140px tall (26.9mm) and 39.8% of the size of the native landscape iPad keyboard.

**ABSTRACT**
Current soft QWERTY keyboards often consume a large portion of the screen space on portable touchscreens. This space consumption can diminish the overall user experience on these devices. In this paper, we present the 1Line keyboard, a soft QWERTY keyboard that is 140 pixels tall (in landscape mode) and 40% of the height of the native iPad QWERTY keyboard. Our keyboard condenses the three rows of keys in the normal QWERTY layout into a single line with eight keys. The sizing of the eight keys is based on users' mental layout of a QWERTY keyboard on an iPad. The system disambiguates the word the user types based on the sequence of keys pressed. The user can use flick gestures to perform backspace and enter, and tap on the bezel below the keyboard to input a space. Through an evaluation, we show that participants are able to quickly learn how to use the 1Line keyboard and type at a rate of over 30 WPM after just five 20-minute typing sessions. Using a keystroke level model, we predict the peak expert text entry rate with the 1Line keyboard to be 66-68 WPM.

**ACM Classification:** H5.2 [Information interfaces and presentation]: User Interfaces – Input devices and strategies

**General terms:** Design, Experimentation, Human Factors

**Keywords:** Text entry, soft keyboard, reduced keyboard, word disambiguation, keystroke level model


*UIST'11*, October 16–19, 2011, Santa Barbara, CA, USA.


## INTRODUCTION

Touchscreens allow the input and output space to be completely overlapped. This removes the need for physical input keys. Unfortunately, to support typing on touchscreens without additional hardware, many touchscreen devices, such as the iPad, implement a soft QWERTY keyboard that consumes a significant amount of screen space.

We conducted a small survey with 50 iPad owners (26 male, 24 female) to understand their user experience. Although survey respondents reported very frequent usage of their iPads and liked their overall experience, they also indicated a desire to minimize the space occupied by the soft QWERTY keyboard. Respondents rated the display size of the iPad with a median of 5 (1: very unsatisfied, 5: very satisfied); however, the rating significantly dropped to a median of 4 for the display size with the keyboard visible (Friedman, $\chi^2_{(1)}=21.2$, $p<.001$). Comments from the survey highlighted a common reason for this drop.

*"The keyboard size is great to type on, but the remaining screen real estate is too small to work on properly. You have to scroll up and down all the time."*

Feedback from survey respondents inspired us to design a keyboard that consumes less space than the current solutions. Motivated by evidence that users rarely want to spend the time needed to learn a new keyboard layout [1, 17], we decided to leverage the QWERTY keyboard layout instead of using different layouts and input modalities. In doing so, we hope that users can adapt their familiarity and existing experiences with physical keyboards for desktops, laptops or other computer devices to our keyboard. The design we developed, called the 1Line keyboard, condenses the three rows of character keys in the QWERTY layout into a single line with eight keys (Figure 1). In this manner,


Figure 2: A storyboard illustrating a user typing *test*. As the user types, a word disambiguation algorithm tries to identify words she wishes to type. If the word she wants is not first in the disambiguation list, she flicks down to select it.

our keyboard is about 40% of the height of the native iPad keyboard (in landscape mode). The sizing of the eight keys is based on users' mental layout of a QWERTY keyboard on an iPad. The system disambiguates the word the user types based on the sequence of the keys pressed and using word frequencies calculated from an English corpus. If her desired word is not the first disambiguated word, she can flick through a list of possible words to select the intended text. This concept is similar to other works that condense multiple letters into one key [4, 5, 6] and list disambiguated words [6, 11, 21]. Our keyboard goes one step further by supporting flick gestures for backspaces and enters, and taps on the bezel below the keyboard for space (Figure 2).

In this paper, we describe our implementation and our user studies and analysis of the 1Line keyboard. Our participants were able to achieve 30.7 WPM (SD: 10.8) on the 1Line keyboard after five 20-minute typing sessions. While this rate is 57% of the participants' rate using the native iPad QWERTY keyboard (iPad: 53.9 WPM (SD: 19.8)), our 1Line keyboard saves 60% of the space occupied by the iPad keyboard. Our study also included the task of creating presentation slides and revealed that participants needed to perform significantly more scrolling with the iPad keyboard than the 1Line keyboard. Finally, we discuss a keystroke level model of our keyboard design which predicts the peak expert text entry rate to be 66.8-68.6 WPM.

## RELATED WORK

The 1Line keyboard builds on prior work in keyboards with reduced layouts and gestures to replace buttons or keys.

### Keyboards with Reduced Layouts

The limited space available in portable devices motivated researchers to reduce the number of buttons required to support a full alphabetic keyboard [4, 5, 6]. These systems must employ word disambiguation techniques to address the fact that a reduced key set requires that each key represent more than one character. Letterwise [14] and the commercially available T9 [21] represent attempts to support text entry on the 12 keys of a numeric keypad on a mobile device. The primary difference between these systems is the word disambiguation strategy: Letterwise orders the possible meanings of a key sequence by prefix probability while T9 orders possible meanings using whole word probabilities. In T9's disambiguation algorithm, 93.5% of the corpus of the top 9025 most frequent English words can be typed uniquely [19], whereas in our algorithm, 94.3% of the top 10,000 words can be typed uniquely.

Brailletouch is a 6-key chording keyboard for text entry [4] but requires knowledge of braille. Green *et al.* introduced the "stick keyboard" which consists of the home keys on the QWERTY keyboard along with a second row of modifier keys [6]. While their design is similar to ours, we reduce the keyset even more by introducing bezel taps and swipe gestures. Furthermore, our layout is based on users' mental layout of a QWERTY keyboard on an iPad.

### Gesture Support to Replace Buttons or Keys

Researchers have previously explored gestures both on and off the screen to replace touch button inputs. The Non-Keyboard is a concept that maps a QWERTY layout into a glove [5]. The Non-Keyboard disambiguates words using three levels: lexical (tri-grams), syntactic (part of speech) and frequency rankings (corpus). However, due to the limited computing power available at the time of the research, the authors evaluated the system in a Wizard of Oz study. The authors were not able to show the performance and learning effects of their concept in a multi-session study. With the 1Line keyboard, we choose to use gestures to augment the input keypad rather than completely eliminate the touch keypad.

Flick and swipe gestures have been used to replace touchscreen buttons [2, 10, 12, 13, 18]. These projects uncover the fact that swipe gestures need not rely on a starting spot on the screen, which lets us reuse screen real-estate for both taps and swipes. Another approach to replace on-screen buttons is to interact with devices by tapping off the screen. Tap detection using accelerometers has been investigated before with Whack gestures [8] and Bonfire [9]. We use tap detection to replace the spacebar. Our 1Line keyboard utilizes commodity hardware (touchscreen and accelerometer) to provide a reduced-size keyboard that leverages prior experiences with the QWERTY layout.

## DESIGN AND IMPLEMENTATION

The basic concept of the 1Line keyboard is to combine each column of the keys in the QWERTY keyboard into one key in order to save space while maintaining the familiar QWERTY layout. Because each key is now associated with multiple letters, this design needs to perform word disambiguation. We also introduce flick gestures to replace backspace and enter keys, and a tapping gesture on the bez-

el below the keyboard to replace the space bar. Figure 2 shows a walkthrough of typing the word *test*. We explain the implementations of these components in this section.

### Key Layout and Size

To inform the design of the 1Line keyboard, we conducted a small study with nine participants (6 males, 3 females) in order to understand the spatial distribution of typing on the QWERTY layout. Findlater *et al.*'s work on touch typing patterns suggests that the user's fingers do not necessary line up in a straight line when typing on the Microsoft Surface [3]. Furthermore, different fingers had different spreads in target selection. We thus decided to empirically identify the optimal key size and layout.

*Setup and Methods.* Similar to the *Asterisk feedback, no keyboard* condition in Findlater *et al.*'s study [3], we asked the participants to type test phrases on a blank keyboard (Figure 3). Two coloured bars were marked on the blank keyboard for participants to home their index fingers, similar to the bars on the *F* and *J* keys on most QWERTY keyboards. The gray bounding box in Figure 3 is of the exact size of the three rows of keys in the native iPad keyboard. To prevent participants' hands from drifting over the course of the study, the space bar was condensed into two small yellow buttons, one for each thumb. Participants had to touch one of these small buttons with the thumb in order to enter a space. Also, to facilitate our analysis of the gathered data, we disabled multi-touch on the device, and participants were instructed to type as accurately as possible.

Participants were required to type 50 phrases. Thirty of the phrases were randomly picked from the MacKenzie & Soukoreff set [16]. In order to mitigate the imbalance on the number of occurrences of letters like *j*, *y*, and *z*, we created 20 special phrases. These 20 phrases were created using the following steps. First, we looked for the most frequent words that use infrequent letters in the Corpus of Contemporary American English[1] (COCA). Then we manually identified meaningful phrases containing these words. For example, one of the phrases *"the grey zone of justice"* contains three infrequent letters: *j*, *y*, and *z*. The final set of 50 phrases had at least 13 appearances of each letter and had a correlation with English of 0.9205 (calculated using AnalysePhrases [16]; first 50 phrases from the MacKenzie & Soukoreff set had a correlation of 0.939).

Participants were given five practice phrases at the beginning of the session. To ensure a consistent starting hand position, users had to press both the space buttons with their thumbs to activate a test phrase. As feedback, we showed an asterisk for a non-space character and an underscore for a space character (see Figure 3). We collected the contact points for every alphabetic character that was not backspaced along with the label of the corresponding letter in the test phrase. This study took around 25 minutes to complete.

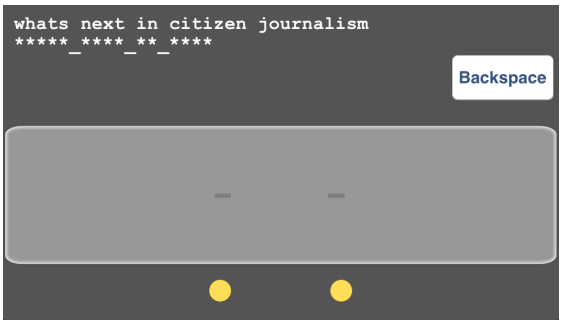


Figure 3: Interface used to record the participants' mental layout of a QWERTY keyboard on an iPad.

*Results.* Contact points for each character beyond 3SD of the respective participant mean were removed as outliers. The remaining points for each character were averaged across all participants and are presented in Figure 4. The error bars in the Figure 4 represent 2SD. Instead of the distinct hand curvatures seen in the results from Findlater *et al.*'s study [3] (keys using the middle and ring fingers were higher than keys using the index and pinky finger), our results show that the middle row of the keyboard is relatively flat. We believe that this is due to the constrained size of the interaction space. In Findlater *et al.*'s study, participants typed on a Microsoft Surface and were not constrained for space. However, in our study, because the input space on the iPad is confined, the participants had to angle their hands slightly inward and curl their fingers closer than normally needed for a full sized keyboard.

*Design of the keys.* Our objective for this study was to find a comfortable size and position for each of the eight keys in the 1Line keyboard. We looked closely at the results of the keys located in the home row. To instruct the height of the keys, we took the distance between the highest and lowest error bars of the home row (*L* and *D* ~120 pixels). To divide the horizontal space into different buttons, we used the midpoint between the ends of adjacent errors bars for the points of *A*, *S*, *D*, and *F* (see Figure 4). These divisions gave a layout for the four left hand keys. We then mirrored the layout to the right half. This symmetry keeps the layout consistent and clean. We based our calculations on the left handed letters (*i.e., A, S, D,* and *F*) instead of the right because the most frequently used letters in the alphabet are located on the left side of the QWERTY keyboard.

### Word Disambiguation

Because each of the eight keys is associated with multiple letters (*e.g.,* the leftmost key can type the letters *q*, *a*, or *z*), word disambiguation is necessary for successful typing. We implemented our algorithm for word disambiguation based on the COCA. We first removed all entries in the COCA containing non-alphabetic characters. We then built a hash table of all remaining words using the finger sequence as the key and the word as the value. For example, the word *test* is assigned to a finger sequence of *left-index, left-*

[1] The version we used contained 500k of the most frequent words used in the American English. Each word is given a frequency as indications of how popular the word is in American English.

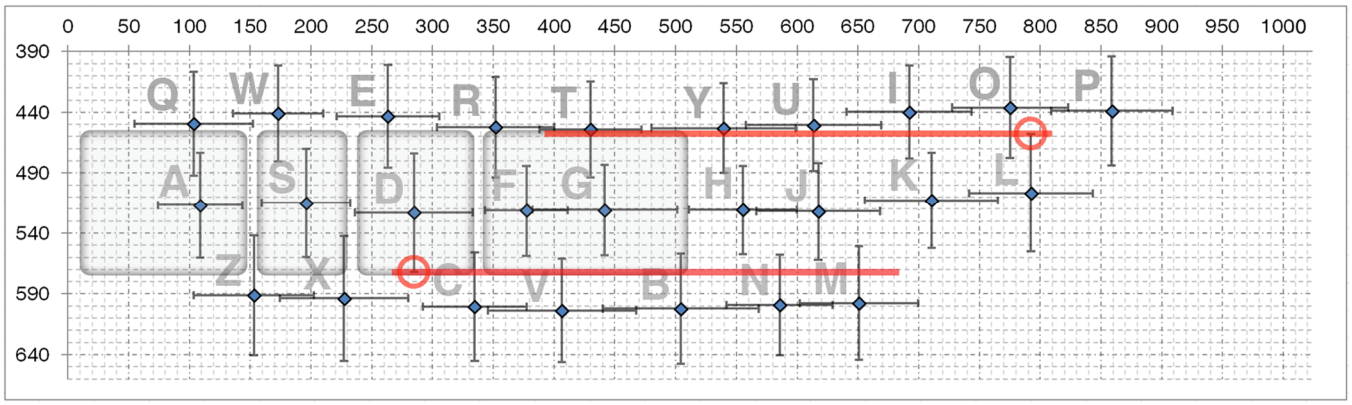


Figure 4: Participants' mental layout of a QWERTY keyboard on an iPad. The units are in pixels. The error bars show 2 SD of the means. The four boxes show how we designed the layout of the buttons in the 1Line keyboard. The height of the buttons was chosen to reflect the highest and lowest error bars of the home row (circled in red). To maintain symmetry we designed the left-handed buttons and mirrored them to the right (only left-handed buttons are shown here).

*middle*, *left-ring*, *and left-index*. We sorted all the words that have the same finger sequence by their frequency in the COCA. Thus, our word disambiguation algorithm suggests the most frequently-used word first depending on the sequence of the key pressed.

*Coverage.* Figure 5 shows the coverage of the alphabetic only words in COCA using our algorithm. To calculate coverage, we found the percentage of words that appear as the most likely word for its key sequence and the percentage of words that appear in the top two and three most likely words for its key sequence. These results show that in the top 10,000 words, 94.28% can be typed without the need for disambiguation and 99.23% appear in the top three most likely words for any key sequence.

### Flick Gestures

We implemented some common operations in text entry as flick gestures. The entire keyboard space is a gesture area. We currently support the following gestures:

- *One-finger left flick*: Backspace (hold to repeat)
- *One-finger right flick*: Enter
- *One-finger down flick*: Navigates to the next word in the word disambiguation list (hold to repeat)
- *One-finger up flick*: Navigates to the previous word in the word disambiguation list (hold to repeat)
- *Two-finger left flick*: Deletes the whole word

### Bezel Tap Detection

Like most mobile devices on the market today, the iPad is packaged with a MEMS accelerometer. For the 1Line keyboard, we took advantage of this existing hardware to detect tapping gestures on the bezel. We sampled the accelerometer at 100 Hz and computed the 2nd order finite difference ($fd_{[n]}$) using the values in the Z-axis (the axis perpendicular to the device screen). Each accelerometer measurement provides the system time ($t_{[n]}$) and the acceleration along the z axis ($z_{[n]}$). The formula for $fd_{[n]}$ is:

$$fd_{[n]} = \frac{z_{[n-2]} - 2z_{[n-1]} + z_n}{(t_{[n-1]} - t_{[n-2]})(t_{[n]} - t_{[n-1]})} \quad (1)$$

Figure 6 shows the $fd_{[n]}$ over the course of two taps. We set a window of ten previous values (equivalent to 100ms) to watch for a significant positive and negative deviation from zero in the finite difference. Empirical evidence suggests 800 m/s$^4$ as an acceptable threshold. The system also monitors all touch events on the screen. When an acceleration spike occurs and a touch-down event does not happen con-

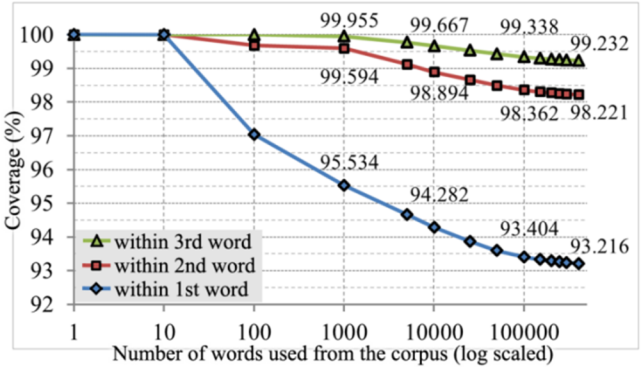


Figure 5: Coverage of our word disambiguation algorithm. Each line shows the coverage of words that are returned within the first, second, or third word of the word disambiguation list. There were a total of 406912 alphabetic only words in the corpus.

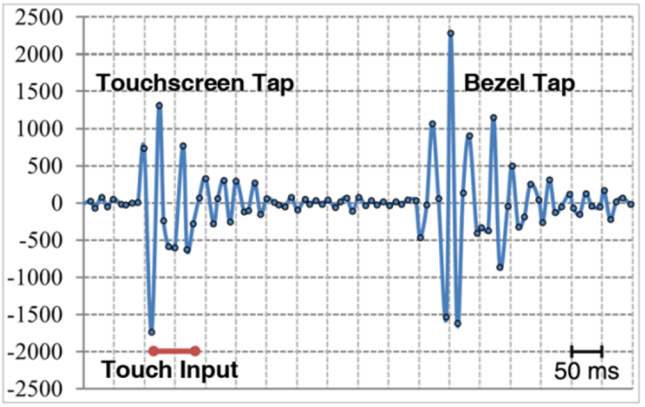


Figure 6: This graph of $fd_{[n]}$ shows a tap on the touchscreen followed by a tap on the bezel. The touch input is labelled in red.

currently, the system considers the spike as a tap on the bezel. When the difference becomes greater than 10,000 $m/s^4$, the system ignores the acceleration data. This can happen when, for instance, the user moves the device. The system ignores acceleration data while the user is touching the screen and again for the first five samples after detecting any tap to avoid false detection of multiple taps. Our algorithm is computationally inexpensive and can keep up with the fast typing speed of expert users.

## USER EVALUATION

In this study, we wanted to validate the usability of the 1Line keyboard and to compare it to a soft QWERTY keyboard. We conducted our study using the Apple iPad because it is currently the most popular tablet on the market. Because of the similarity to the QWERTY layout, we expected the learning curve to level fairly quickly. Our pilot studies confirm that although typing speed still continues to improve after 15 sessions of typing, it diminishes significantly after the first 5 sessions. Thus, we decided to format the study into 5+2 sessions. In the following section, we will refer to these 5+2 sessions as the five *typing sessions*, an *application session* and the *spacebar session*.

The *typing sessions* were used to gain an understanding of the initial usability of the 1Line keyboard and its learning curve. The *application session* was designed to collect both qualitative and quantitative data of the participants' experience with using the 1Line keyboard to make presentation slides. In our exploratory survey, creating presentation slides was the least preferred activity on the iPad by the responders. The inclusion of this session in the experimental design allows us to understand how the 1Line keyboard could benefit users in a realistic scenario in which the limited visual working space can cause interaction breakdowns that affect the ease in which the user completes a task. The *spacebar session* was designed to evaluate our bezel tap detection.

### Participants

Ten participants (referred to as *P0-P9;* 8 males; 8 right handed) from age 20 to 35 (mean: 23.7, SD: 4.4) were recruited for our study. All but one participant had at least one touchscreen device (mobile phones and/or music players) and regularly engaged in text entry on these devices. However, none had significant prior experience typing on the native iPad keyboard. All participants showed expert typing speed on a physical QWERTY keyboard using TextTest [22]; the average typing speed was 82.6 WPM (SD: 23.0) with about 4% error rate. They were compensated $120 for their participation in this study.

### Instruments and Setup

We used a 9.7" 16GB iPad. The iPad was placed on a desk in front of the participants, who could choose to prop the iPad up to any angle they felt comfortable. They could also freely adjust the height of their chairs.

*The Typing Sessions.* The *typing sessions* used a custom iPad application. The application presented short test phrases above either the 1Line keyboard or the iPad keyboard. Aside from the keyboards, the participants were not able to interact with any other part of the screen. We presented the iPad keyboard in the default factory state with the auto-correction enabled. The phrases used in the typing tasks are from the set of 500 test phrases developed by MacKenzie & Soukoreff [16]. We randomized the order of these phrases and added them to a list. We continued this until we had 5000 phrases in two lists, one for each keyboard. We then grouped every ten phrases into a block, making 500 blocks for each list. These two sets of blocks were pre-loaded into the iPad application. The application kept track of the progress for each participant. When a participant started a new session, the system always presented the first phrase in the next block even if she did not complete all the phrases in the previous block during the previous session.

*The Application Session.* In the *application session*, the participants were asked to complete presentation slides on our simplified presentation application. Figure 7 shows a screenshot of our presentation application. The design of this application was based on Keynote, which is an existing presentation application on the iPad. Aside from interacting with the keyboard, participants could pan the slide up and down to view the portion of the slide occluded by the keyboard as well as minimize the keyboard entirely by tapping on the slide. Tapping the title bar reactivated the keyboard.

*The Spacebar Session.* The app used in the *spacebar session* was very similar to the one used in the *typing sessions*. However, we disabled the bezel tap detection and the single row of keys was shifted up to accommodate a touchscreen spacebar. The spacebar had the exact same height as the bezel (117 pixels tall, ~22mm) and extend from the *left-middle* finger key to *right-middle* finger key.

### Study Design and Procedure

*The Typing Sessions.* In each of the *typing sessions*, participants used both the 1Line and the iPad keyboard to type the test phrases. In each session, participants completed 20 minutes of typing with each keyboard. Each time partici-


Figure 7: The Keynote imitation app used in the *application session* with the native iPad keyboard (left) and 1Line keyboard (right).

pants finished a 20-minute typing condition, they were asked to complete a NASA TLX workload assessment [7]. The presentation order of the two keyboards was counter-balanced across participants in the first sessions and alternated in each session thereafter. Each session took around 45 minutes to complete. Any two of the sessions were scheduled at least 2 and at most 72 hours apart so that participants could carry over their experience on the 1Line keyboard throughout the study.

*The Application Session.* Participants in this session had to add the title to a series of slides for a real-estate presentation. Each slide contained three images: a map, a floor plan, and a picture showing interior images from a house. Bullet points above the floor plan described what characteristics of the house to put in the title. Each of these points required the participants to look at the one of the three images. Participants had to follow the order given in the bullet points. For example, for the slide shown in Figure 7, they would have to type *"near school one kitchen two bathrooms corner of county and meadow tile flooring three bedrooms"*. There were seven slides in total: one for practice and three for each keyboard. The presentation order of the keyboards was counter-balanced across participants. Participants were given as much time as they needed to complete the tasks and were allowed to stop and ask questions at any time. A short interview was conducted right after to probe their experiences with the two keyboards in this application. They were encouraged to make comments in the context of the presentation titling tasks.

*The Spacebar Session.* Participants carried out another 20-minute typing exercise using a slightly modified 1Line keyboard. In this session, we included a spacebar in the touchscreen and disabled the bezel tap detection. We administered this extra typing session to check whether our bezel tap detection could have degraded the typing speed. After the typing, we administered a workload assessment and conducted a final exit interview. The *application session* and this *spacebar session* were scheduled together and took 45 minutes to complete in total.

**Results: Typing Sessions**

For our analysis, we modified the StreamAnalyzer from Wobbrock and Myers [23] to account for word-level operations like word deletion and word corrections. Also the native keyboard's auto-correct function can potentially lower the keystroke per character (KSPC) below 1.0. We accounted for these cases in our analysis.

*Overall Performance.* We removed any data point over 3SD from the average typing rate and over 4SD from the average error rate for each participant in each session as an outlier (0.95% of the data points removed). Running a repeated-measures ANOVA for each keyboard against the sessions, we found a significant main effect of the number of the sessions on typing speed (1Line: $F_{(4,36)}$=30.82, $p$<.001, iPad: $F_{(1.937,17.434)}$=4.54, $p$=.027 with Greenhouse-Geisser correction $\varepsilon$ =.48). However, the post-hoc pairwise comparisons show no significant differences between sessions 3 to 5 for each of the techniques (1Line: $p$>.64 iPad: $p$>.99 for all pairs). Because both the QWERTY keyboard and the 1Line keyboard use the QWERTY layout, we expected a shallow learning curve. Moreover, we found no significant main effect of session on the differences in speed between the two techniques ($F_{(4,36)}$=7.35, $p$=.80). In other words, participants displayed a similar learning curve as the iPad keyboard, which affirmed that they were able to transfer their existing experiences from using QWERTY keyboards in desktops, laptops and other devices.

With paired t-tests, the 1Line keyboard performed significantly slower ($t_{(9)}$=7.74, $p$<.001) than the iPad keyboard in each of the 5 sessions (see Figure 8 left). At the 5th session, participants averaged 30.7 WPM (SD: 10.8) on the 1Line keyboard and 53.9 WPM (SD: 19.8) on the iPad keyboard. They were able to reach 57% of the performance on the iPad keyboard using a keyboard that is 60% smaller.

The NASA TLX workload assessments were analyzed with paired t-tests (the normality of mental and total workloads in each session passed the Shapiro-Wilk tests). We found no significant differences between the 1Line and iPad keyboards in any of the 5 sessions at the 95% confidence level. Looking at individual components of the assessments, we only found a trend in mental demand (see Figure 8 right). The differences in mental demand between the two keyboards in the first four sessions was significant ($p$<.036 for all sessions). But at the 5th session, the differences were not significant ($t_{(9)}$=2.09, $p$=.066). While our participants had more mental demand when typing on the 1Line keyboard compared to the iPad keyboard, the cognitive effort became comparable after the 5th session.

*Error Rates.* The final error rate [22] is computed using the minimum string distance (MSD) [20]. Participants had a final error rate of 1.7% (SD: 2.4) with the 1Line keyboard and 1.2% (SD: 2.0) with the iPad keyboard. A Wilcoxon signed ranks test showed no significant difference on the final error rate between the 1Line keyboard and the native iPad keyboard ($Z$=-0.87, $p$=.386). On average, the KSPC for the 1Line keyboard was 1.31 (SD: 0.35) and the iPad key-

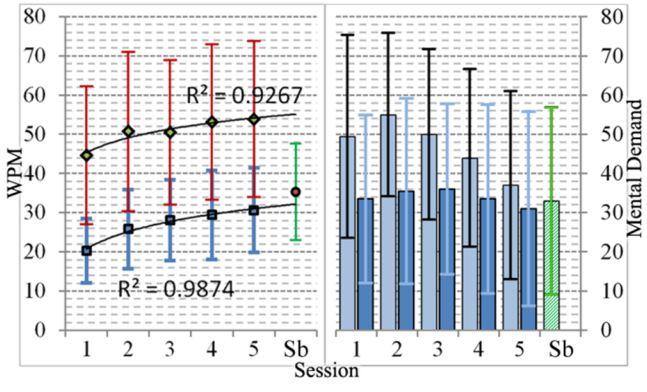


Figure 8: Left shows the overall text entry rate for both keyboards (iPad: top, 1Line: bottom) for each session. Right shows the mental demand for both keyboards (iPad: dark, 1Line: light) in each session.

board was 1.26 (SD: 0.36). Pairwise t-tests show significant differences in KSPC between the two keyboards in each of sessions 3 to 5 ($p<.05$ for all pairs). This is not surprising once we look at the corrected error rate. On average, participants had a corrected error rate of 10.8% (SD: 10.8%) with the 1Line keyboard and 9.0% (SD: 10.6%) with the iPad keyboard. A Wilcoxon test showed a significant difference in the corrected error rate [20] between the two keyboards ($Z$=-2.40, $p$=.017). There were fewer manual corrections with the iPad keyboard and, hence, a lower KSPC. This is because the factory auto-correct function was enabled for the native iPad keyboard condition. On average the iPad automatically corrected 7.5% (SD: 10.1%) of the input stream. This auto-correction rate translates to one correction every 13.3 characters or roughly once every 3 words.

#### Results: Application Session

Because participants could type faster on the iPad keyboard, we found a significant difference in completion time between the keyboards ($t_{(9)}$=-5.45, $p<.001$). Participants used on average 105.6s (SD: 26.0) with the 1Line keyboard and 77.8s (SD: 20.2) with the iPad keyboard to title each slide. The 35.7% slowdown with using the 1Line keyboard in this task almost directly reflects the 43% slowdown observed in the *typing sessions*.

However, our main objective of this task was to measure how likely the user would need to perform interactions on the visual working space (*e.g.*, scrolling) to gain information to complete the task. The average distance scrolled while completing the tasks for each slide was 1325 pixels (SD: 846) using the 1Line keyboard and 2828 pixels (SD: 1819) using the iPad keyboard. The distances scrolled between the two techniques are significantly different ($t_{(9)}$=3.84, $p$=.004). This result indicates that a larger visual working space offered by the 1Line keyboard allowed participants to focus on completing the task rather than navigating the working space.

#### Results: Spacebar Session

In this last session, we added a soft spacebar. On average, participants achieved 35.3 WPM (SD: 12.3). The data from the first five sessions fits a logarithmic growth ($R^2$=0.9874, Figure 8 left). Using the fit, the text entry speed can be extrapolated to 32.2 WPM for the next session. Thus, our bezel spacebar only degraded performance by about 8.8% and saves the space used by the soft spacebar (117 pixels tall or ~22mm in our implementation).

#### Discussion on Performance Differences

At the fifth session, our participants performed 43% slower on the 1Line keyboard than the iPad keyboard. Aside from the fundamental trade-off between keyboard size and performance (1Line keyboard is 60% smaller than the iPad keyboard), we examined the results and found three main sources for the performance drop.

First, the inaccuracies of the bezel spacebar detection algorithm caused some false positives and false negatives. The results of the *spacebar session* showed that the inaccuracies attributed to 8.8% of the performance degrade. Furthermore, we estimated the frequency of false positives by counting space characters that were backspaced. On average 5.2 (SD: 1.1) space characters were backspaced per block (10 lines). Thus, a false positive is estimated to have occurred roughly once every 2 lines. The bezel spacebar also had false negatives; however, we were not able to quantify the amount of false negative with the data we collected.

Second, with disambiguation, it was difficult to catch a typo in the beginning of a word (*e.g.,* a mistype of finger, *fonger*, disambiguates to *bomber*). Five participants said they often needed to delete the entire word to start over. On average, participants used the word backspace function 2.7 (SD: 1.0) times per block. The phrase set we used contained an average of 64 words per block. Participants thus threw out about 4.2% of all words they typed.

Lastly, the current implementation does not remember the word the user selected in the disambiguation list. If a wrong word is selected by accident, the user would need to backspace and start from the very top of the list again. Four participants raised this concern during the interview. Further analyses showed that participants used an average of 3.5% (SD: 6.0) of their typing time interacting with the list.

#### Qualitative Results of User Preference

All participants but one thought they were faster with the iPad keyboard. But in the context of making presentation slides, five explicitly said they needed to scroll less with the 1Line keyboard.

*"I spent a lot of time going back and forth on the slides for the iPad keyboard, like going up and down, because the keyboard takes up half the screen." (P1)*

After the slides, participants even began to envision other scenarios where the 1Line keyboard would be useful.

*"When we compare it with the other keyboard, well I can say that it just holds a big portion of the screen. That's maybe a bit disturbing if you are typing a long piece of document, instead of just single sentences." (P4)*

While the entire size of the 1Line keyboard is 40% of the size of the iPad keyboard, the individual keys are actually much larger. Our participants found the larger keys more comfortable to touch.

*"The [1Line keyboard], it's really convenient, because I can just place my hands like this. And basically, I don't have to move them a lot. I just move my fingers. I find this more comfortable than the regular one. I sometimes have to get my hand out of the way to see some of the keys on the normal one." (P7)*

Another advantage we found with the keys of the 1Line keyboard is that it facilitates typing with long nails.

*"And like when I curl my fingers for the lower keys, my nail hit and then I don't touch the [iPad] keyboard or the screen and then I don't touch anything, which doesn't happen with the [1Line] one." (P1)*

The 1Line keyboard reduces the screen space needed while maintaining the QWERTY layout. This liberates the effort needed for users to learn a completely new layout.

*"I was still typing sort of like a smaller version of regular typing. So if it's a top row key, my fingers will actually open up a little…" (P1)*

The disambiguation list was the least preferred aspect of the 1Line keyboard.

*"I find it difficult in doing sort of longer words. I find it difficult trying to keep my position inline where I was in the word. […] So I found myself sort of having to spell the word letter by letter in my head as I was typing. So it required a bit more mental effort." (P0)*

Because the system does not support word prediction, the target word might not appear in the disambiguation list until the final character has been typed. This might have caused extra mental effort in our participants. Our participants also expressed difficulty in cases where the same key needs to be pressed multiple times (*e.g.*, the word "indeed" requires users use the left middle finger four times).

*"On the single row one, when I am typing sometimes, when I have to press one of the keys too many times, I usually make mistakes there." (P7)*

Another major difference between the 1Line and the iPad keyboard was the flick gestures for backspace and enter. Participants adapted to these gestures without troubles. Four participants mentioned swipe gestures when asked what they liked in the 1Line keyboard.

*"The swiping, I thought that was really cool. It really makes it easier. It's more like a reflex rather than I have to go and find another key. Because when you are tapping you just want to worry about the letters, nothing else. You don't want to use another key for enter or backspace." (P8)*

## PREDICTING PEAK EXPERT PERFORMANCE

The peak expert performance is another metric to understand a new text entry method. Instead of extending our user study, we predict the expert performance of the 1Line keyboard using a keystroke level model (KLM). A KLM is built from the times of individual keystrokes. But because there are no existing time measurements for multi-finger touchscreen keystrokes, we ran an additional study with the same participants as our user study to collect keystroke and flick gesture timing data for building the KLM. This study took place right after the last session of the user evaluation.

We did not build a similar KLM for the iPad keyboard because we would need to measure $(26 \text{ letters} + 2 \text{ thumbs})^2 = 784$ transitions. Such a study would be too fatiguing for our participants. Because our main contribution is not proposing models of different keyboards, we decided not to pursue a model for the iPad keyboard in this work.

### Keystroke Level Model

Consider typing the word *wpm*. The first letter *w* is typed with the *left-ring* finger. After typing *w*, the user will hit the letter *p* with the *right-pinky* finger. This sequence of typing two letters then consists of three elements: pressing the *left-ring* key, transitioning from the *left-ring* key to the *right-pinky* key, and pressing the *right-pinky* key. Thus, the time to type *wp* can be modeled as the sum of these two key press times and one transition time. Even after the user types the whole word, she may need to select a word from the disambiguated list. For this, she would perform downward flick gesture(s) to reach the desired word. In the example of *wpm*, it appears third in the list. Thus, the user has to perform a flick gesture twice. Our KLM takes the flick gesture and transition times into account.

In typical typing situations, every word is followed by a space. Our KLM assumes that every word ends with a space and starts from a space. A space character can be entered by either the *right-* or *left-thumb*. Depending on the last keystroke of the word, the time for entering a space character with the *right-thumb* or the *left-thumb* will be different. In the example of *wpm(flick)(flick)*, the last interaction before entering a space is a flick gesture. The transition time from a flick gesture to the *right-thumb* can be different from one to the *left-thumb*. The key press time can also be different between the two thumbs. Similarly, because we assume every word starts from a space, our KLM needs to add the transition time from a space to the first keystroke of the word. We generated two models; the *ideal thumb policy* KLM and the *non-ideal thumb policy* KLM. The *ideal thumb policy* KLM uses the fastest times to account for the space characters. On the other hand, the *non-ideal thumb policy* KLM uses the slowest times.

### Experimental Setup and Methods

We developed an iPad application to measure the timing of keystrokes on the 1Line keyboard. The bottom portion of the screen showed the 1Line keyboard with unlabelled keys. We added two spacebars below the buttons for each of the thumbs and disabled the tap detection algorithm. The top portion displayed a disabled copy of the bottom portion and was used to present the tasks to the participants.

Each task consisted of three consecutive interaction steps. An interaction step can be either tapping a key or performing a flick gesture. For example, one of the tasks consisted of tapping the *right-middle* finger button, flicking down, and then tapping the *left-thumb* button. These interaction steps were illustrated at the top of the screen. Because multi-touch was enabled, participants could begin the next interaction step (*i.e.,* putting a finger down on the screen) without completing the previous interaction step (*i.e.,* lifting a finger up on the screen) as long as they finished the previous interaction step before finishing the next interaction. Each task had to be completed correctly; otherwise, participants were asked to complete the same task again. Participants were instructed to only use the corresponding finger for each key (but they were allowed to perform flick gestures with any finger) and to complete each task as fast as they could. We collected the key press times for all keys, the transition times for all possible pairing of keys and flick gestures, and the times for flick gestures. Unreachable transitions were not collected (*e.g.,* transition for a space to a space). This experimental design allowed us to measure the time for different interaction components without a mental

| | *l-pinky* | *l-ring* | *l-middle* | *l-index* | *l-thumb* | *r-thumb* | *r-index* | *r-middle* | *r-ring* | *r-pinky* | *flick UP* | *flick DN* |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *l-p* | **86**±30 | **71**±87 | **72**±115 | **56**±97 | **81**±151 | **57**±93 | **69**±146 | **74**±94 | **61**±87 | **52**±92 | - | - |
| *l-r* | **64**±83 | **87**±43 | **64**±84 | **62**±102 | **91**±141 | **75**±127 | **89**±126 | **95**±134 | **69**±117 | **66**±100 | - | - |
| *l-m* | **83**±123 | **72**±130 | **92**±33 | **52**±69 | **87**±121 | **83**±126 | **112**±164 | **88**±139 | **104**±137 | **72**±96 | - | - |
| *l-i* | **83**±109 | **79**±96 | **52**±77 | **99**±52 | **94**±111 | **85**±112 | **63**±80 | **95**±150 | **106**±161 | **67**±87 | - | - |
| *l-t* | **59**±85 | **80**±88 | **70**±93 | **74**±90 | - | - | **67**±90 | **84**±158 | **78**±140 | **74**±99 | **232**±105 | **224**±124 |
| *r-t* | **61**±113 | **83**±120 | **93**±137 | **70**±102 | - | - | **50**±82 | **67**±98 | **80**±130 | **86**±101 | **248**±106 | **229**±107 |
| *r-i* | **76**±122 | **96**±138 | **85**±115 | **62**±98 | **67**±96 | **73**±117 | **91**±33 | **53**±74 | **62**±88 | **70**±113 | - | - |
| *r-m* | **76**±146 | **77**±90 | **85**±133 | **89**±108 | **69**±102 | **81**±114 | **69**±91 | **94**±53 | **58**±93 | **95**±107 | - | - |
| *r-r* | **78**±122 | **61**±91 | **78**±130 | **92**±135 | **59**±93 | **83**±121 | **88**±155 | **66**±88 | **91**±39 | **81**±138 | - | - |
| *r-p* | **56**±102 | **71**±132 | **75**±114 | **79**±113 | **61**±88 | **76**±121 | **77**±162 | **86**±160 | **93**±131 | **92**±50 | - | - |
| *UP* | **271**±284 | **230**±161 | **235**±185 | **200**±116 | - | - | **213**±131 | **336**±334 | **315**±313 | **295**±241 | **158**±82 | - |
| *DN* | **255**±196 | **237**±163 | **243**±222 | **200**±115 | - | - | **248**±237 | **276**±212 | **270**±191 | **254**±154 | - | **143**±103 |
| *KP* | **108**±37 | **102**±33 | **91**±30 | **83**±26 | **101**±28 | **92**±30 | **85**±27 | **92**±28 | **94**±31 | **99**±33 | **108**±42 | **111**±46 |

Table 1: The units are in ms (mean: bolded, SD: unbolded). The bottom row shows the average times of different keystrokes. The top rows show the transitions times from a keystroke to another; columns are the starting keystroke. For example the transition time from *left-ring* to *left-pinky* is 71±87 ms, but the time from *left-pinky* to *left-ring* is 64±83 ms.

overhead. We therefore believe that this performance could be used to predict the peak performance of an expert user.

### Results

Outliers beyond 4SD of the means in each category were removed per participant (1.0% removed). Table 1 shows the resulting means. To come up with the predicted WPM, we followed a method used by MacKenzie & Soukoreff [15]. We first calculated the time to type each word in COCA. Summing up all words, our model used $3.89\times10^8$ (*non-ideal thumb policy* KLM) to $4.00\times10^8$ (*ideal thumb policy* KLM) seconds to type the entire corpus (weighted by frequency). There are $2.23\times10^9$ characters in the corpus (weighted by frequency). The average time required to type a character in the corpus ($t_{CHAR}$) is then 0.17 to 0.18 *seconds per character*. Finally, words per minute (WPM) is calculated by $1/t_{CHAR} \times 60$ *seconds per minute* / 5 *characters per word*. The resulting predicted peak expert performance of the 1Line keyboard is 66.8 (*non-ideal thumb policy* KLM) to 68.6 (*ideal thumb policy* KLM) WPM.

This prediction does not include using the bezel spacebar. While there is a degradation of performance when using the bezel spacebar, our user evaluation showed that the lost in performance was 8.8%. Thus, the estimated peak expert performance with the bezel spacebar would be around 61-63 WPM. Referencing back to the results of the *typing sessions*, the fastest line completed by our fastest participant using the 1Line keyboard was done with 73.8 WPM, the fastest block (10 lines) was done with 48.0 WPM, and the fastest 20-minute session (12 blocks) was done with 40.6 WPM. Our KLM is in line with actual user data.

## AREAS OF IMPROVEMENTS

The current prototype on the iPad was the first step in realizing the concept of the 1Line keyboard. In this section, we discuss a few areas of improvements.

### Adaptive Word Disambiguation

Our current disambiguation approach simply uses the frequency of occurrence of each word appeared in an English corpus (COCA in this work). Adaptively changing the behaviour of the word disambiguation could improve the typing speed. For example, if the user types a lower frequency word (according to COCA) repeatedly, 1Line could give that word higher priority in the disambiguation. There also needs to be ways for the user to enter her own words (those not found in COCA) into the algorithm.

### Integration of Auto-correction

The current 1Line keyboard prototype does not have auto-correction. One problem participants expressed about our keyboard was that when they needed to press the same key a few times in order to type correctly, their typing resulted in an error. Thus, among different auto-correction features, omission correction could greatly contribute to improvements on the user experience on the 1Line keyboard.

### Richer Gesture Set

The current prototype only supports five flick gestures. But the underlying hardware supports other types of gestures, and they could be used to enrich our keyboard. In future implementations, the user could toggle the caps-lock by flicking up or down with two fingers. The user also could switch to a numeric or symbolic layout with 3-finger flicks upward or downward. Future work will investigate an efficient layout for numbers and symbols.

### More Robust Tap Detection

The tap detection algorithm we employ rests on the hypothesis that a user's finger press moves the device. As deployed, with the device resting on its bevelled back casing, any movement that is not perfectly centered on the device causes the device to pivot and register a tap. Future work will investigate whether other configurations, including devices whose case is not bevelled, are good candidates for an accelerometer based tap detection approach. We also found that elevating the device on a soft surface, such as a rubberized case, will lower false positives.

Taps that are not along the central plane of the device register a change in the acceleration along the X and, to a lesser extent, the Y-axis. Our preliminary work suggests that movement in the X-axis can be used to distinguish *left*, *right* and *center* bezel taps, which enables a richer keyboard experience without the expense of extra screen real

estate or smaller keys. For example, users could tap the center of the device to enter a space, but tap its left side to toggle caps-lock and the right side to toggle num-lock.

## CONCLUSIONS AND FUTURE WORK

The objective of this work is to reduce the size of touchscreen tablet keyboards. The result of our work is the 1Line keyboard. It is 140 pixels tall and 39.8% the height of the native keyboard found on an iPad in the landscape mode. The 1Line keyboard has only 8 keys, one for each finger. The key sizes are designed based on participants' mental layout of a QWERTY keyboard on an iPad. All character operation keys are replaced by multitouch flick gestures. The spacebar is integrated into the bezel of the device with the support of a novel tap detection algorithm. Our user study indicates that the 1Line keyboard can reduce interaction breakdowns in tasks involving both visual exploration and text entry in portable tablet devices.

We plan to extend this 1Line keyboard to other types of devices using touchscreens. For example, an adaptation of the 1Line keyboard to a handheld touchscreen device could be beneficial because the real estate of the screen in these devices is even more limited than a tablet device. We also believe that the 1Line keyboard could offer some benefits even in a tabletop surface. In this type of device, visual space would not be an issue. However, as our investigation indicates, our system would allow the user to type accurately as long as it detects which finger is used for typing each letter instead of which key is pressed. Thus, the 1Line keyboard on a tabletop surface could liberate the user from concerns about hitting the wrong keys or homing the hand to a generic keyboard layout given by the system.

## REFERENCES


1. Bi, X., Smith, B. A., and Zhai, S. 2010. Quasi-qwerty soft keyboard optimization. In *Proc. of CHI '10*. ACM, NY, USA, 283-286.
2. Bonner, M.N., Brudvik, J.T., Abowd, G.D. and Edwards, W.K. 2010. No-Look Notes: Accessible Eyes-Free Multi-touch Text Entry. In *Proc of Pervasive 2010*. Springer-Verlag, Berlin, Germany, 409-426.
3. Findlater, L., Wobbrock, J., Wigdor, D. 2011. Typing on flat glass: Examining ten-finger expert typing patterns on touch surfaces. In *Proc. of CHI '11*. ACM, NY, USA, 2453-2462.
4. Frey, B., Southern, C., and Romero, M. 2011. BrailleTouch: Mobile Texting for the Visually Impaired. In *Proc. HCII'11*. To appear.
5. Goldstein, M., Book, R., Alsiö, G., and Tessa, S. 1999. Non-keyboard QWERTY touch typing: a portable input interface for the mobile user. In *Proc. of CHI '99*. ACM, NY, USA, 32-39.
6. Green, H., Kruger, J., Faldu, C., and St. Amant, R. 2004. A Reduced QWERTY Keyboard for Mobile Text Entry. In *Proc. of CHI '04*. ACM, NY, USA, 1429-1432.
7. Hart, S. and Staveland, L. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Human Mental Workload, P. A. Hancock and N. Meshkati, Eds.*, North-Holland: Elsevier Science, 139-183.
8. Hudson, S.E., Harrison, C., Harrison, B.L., and LaMarca, A. 2010. Whack gestures: inexact and inattentive interaction with mobile devices. In *Proc. of TEI '10*. ACM, NY, USA, 109-112.
9. Kane, S.K., Avrahami, D., Wobbrock, J.O., Harrison, B., Rea, A.D., Philipose, M., and LaMarca, A. 2009. Bonfire: a nomadic system for hybrid laptop-tabletop interaction. In *Proc. of UIST '09*. ACM, NY, USA, 129-138.
10. Kane, S.K., Bigham, J.P., and Wobbrock, J.O. 2008. Slide rule: making mobile touch screens accessible to blind people using multi-touch interaction techniques. In *Proc. of Assets '08*. ACM, NY, USA, 73-80.
11. Kane, S.K., Wobbrock J.O., Harniss, M., and Johnson, K.L. 2008. TrueKeys: identifying and correcting typing errors for people with motor impairments. In *Proc. of IUI'08*. ACM, NY, USA, 349-352.
12. Karlson, A.K., Bederson, B.B., and SanGiovanni, J. 2005. AppLens and launchTile: two designs for one-handed thumb use on small devices. In *Proc. of CHI '05*. ACM, NY, USA, 201-210.
13. Kristensson, P. and Zhai, S. 2005. Relaxing stylus typing precision by geometric pattern matching. In *Proc. of IUI'05*. ACM, NY, USA, 151-158.
14. MacKenzie, I.S., Kober, H., Smith, D., Jones, T., Skepner, E. 2001. LetterWise: Prefix-based Disambiguation for Mobile Text Input. In *Proc. of UIST '01*. ACM, NY, USA, 111-120.
15. MacKenzie, I.S. and Soukoreff, R.W. 2002. A model of two-thumb text entry. In *Proc. of GI'02*. 117-124.
16. MacKenzie, I.S. and Soukoreff, R.W. 2003. Phrase sets for evaluating text entry techniques. In *Extended Abstracts of CHI '03*. ACM, NY, USA, 754-755.
17. MacKenzie, I.S., Zhang, S.X., and Soukoreff, R.W. Text entry using soft keyboards, *Behaviour & Information Technology, 18* (1999), 235-244.
18. Pirhonen, A., Brewster, S., and Holguin, C. 2002. Gestural and audio metaphors as a means of control for mobile devices. In *Proc. of CHI '02*. ACM, NY, USA, 291-298.
19. Silfverberg, M., MacKenzie, I. S., & Korhonen, P. 2000. Predicting text entry speed on mobile phones. In *Proc. of CHI'00*. ACM, NY, USA, 9-16.
20. Soukoreff, R.W. and MacKenzie, I.S. 2003. Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric. In *Proc. of CHI '03*. ACM, NY, USA, 113-120.
21. T9. http://www.t9.com/
22. Tinwala, H. and MacKenzie, I.S. 2010. Eyes-free text entry with error correction on touchscreen mobile devices. In *Proc. of NordiCHI '10*. ACM, NY, USA, 511-520.
23. Wobbrock, J.O. and Myers, B.A. 2006. Analyzing the input stream for character- level errors in unconstrained text entry evaluations. *ACM Trans. Comput.-Hum. Interact.* 13, 4 (December 2006), 458-489.