

Beyond QWERTY: Augmenting Touch-Screen Keyboards with Multi-Touch Gestures for Non-Alphanumeric Input

Leah Findlater^{1,2}, Ben Q. Lee² and Jacob O. Wobbrock²

¹College of Information Studies
University of Maryland, College Park, MD
leahkf@umd.edu

²The Information School | DUB Group
University of Washington, Seattle, WA
{ benlee90, wobbrock }@uw.edu

ABSTRACT

Although many techniques have been proposed to improve text input on touch screens, the vast majority of this research ignores non-alphanumeric input (*i.e.*, punctuation, symbols, and modifiers). To support this input, widely adopted commercial touch-screen interfaces require mode switches to alternate keyboard layouts for most punctuation and symbols. Our approach is to *augment* existing ten-finger QWERTY keyboards with multi-touch gestural input that can exist as a complement to the moded-keyboard approach. To inform our design, we conducted a study to elicit user-defined gestures from 20 participants. The final gesture set includes both multi-touch and single-touch gestures for commonly used non-alphanumeric text input. We implemented and conducted a preliminary evaluation of a touch-screen keyboard augmented with this technique. Findings show that using gestures for non-alphanumeric input is no slower than using keys, and that users strongly prefer gestures to a moded-keyboard interface.

Author Keywords: Touch screen; text input; gestures

ACM Classification Keywords: H.5.2 [Information interfaces and presentation]: User interfaces - *input devices & strategies*;

INTRODUCTION

For text input on touch screens, the traditional QWERTY keyboard and geometric techniques based on it (*e.g.*, [4]) have attained the widest adoption. While these techniques focus on entering letters and full words, operations involving punctuation symbols or modifiers can be laborious. On most commercial touch-screen keyboards, for example, accessing such features requires a mode-switch to an entirely different key assignment or layout. Moreover, even though punctuation symbols such as ‘.’ are more common than some letters (*e.g.*, *v*, *k*, *j*) [10], the vast majority of touch-screen text entry research only addresses alphanumeric characters. With the increased use of text and instant messaging, punctuation symbols become even more important (*e.g.*, ‘#’ and ‘@’ on Twitter).

We propose a multi-touch gestural approach to augment traditional ten-finger touch-screen keyboards for non-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI '12, May 5–10, 2012, Austin, Texas, USA.

Copyright 2012 ACM 978-1-4503-1015-4/12/05...\$10.00.

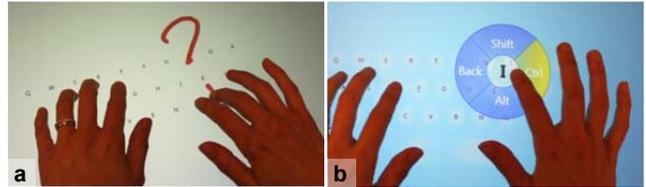


Figure 1. (a) Our multi-touch gesture technique, showing the end of the “?” gesture with all left fingers down and one right finger down; (b) the popup pie menu for modifier keys.

alphanumeric input (Figure 1a): the user places 4+ fingers down with one hand and gestures atop the keyboard itself with the other hand. This bimanual interaction uses active rather than passive modes, supports input with low visual attention, and does not require users to move their hands out of typing position. We envision that these gestures will *complement* rather than replace the existing mode-switched keyboard approach that is the standard for such input.

To create our gestures, we first conducted a study to elicit a set of guessable, user-defined gestures from 20 participants (based on [8,9]). The final set includes single-touch and multi-touch gestures. We then built a gesture-augmented QWERTY keyboard that incorporates the gestures and explores pie menus for those non-alphanumeric characters, such as modifier keys (Figure 1b), that were not found to be as suitable for freeform gestures. Our evaluation of the interface with 6 new participants suggests that subjective responses are positive and that, even for novice users, gestural input does *not* result in a loss in performance.

Our work is inspired by previous research in gestural text input and by augmented keyboard techniques. Although stylus-based text input is not as popular as it once was, several stylus-based techniques have included punctuation gestures, such as Palm OS’s Graffiti (see [6]). However, these techniques differ considerably from our work in that they are designed for single-stroke input with a stylus rather than a finger (or fingers) and support only a single point of input rather than multi-touch. With multi-touch input devices, intuitive, efficient gestures may differ, such as using two fingers instead of one for a quotation mark. Touch-screen keyboard layouts have also been augmented with popup menus, for example, to enter multiple characters in a single stroke [3] and for diacritical marks (*e.g.*, Google’s Android OS and Apple’s iOS). Finally, different levels of pressure have also been used to input lowercase versus uppercase letters, augmenting traditional input [2].

The primary contributions of this paper are: (1) a set of user-defined, guessable gestures for non-alphanumeric text

entry that includes single-touch and multi-touch components; (2) a multi-touch, bimanual input technique providing access to this gesture set; and (3) an evaluation demonstrating that using gestures for non-alphanumeric input is no slower than using keys, and that users strongly prefer gestures to a moded-keyboard interface.

USER-DEFINED NON-ALPHANUMERIC GESTURES

We conducted a study to elicit a guessable gesture set that includes 22 *punctuation symbols* (e.g., ‘: #’) and 4 *commands* (Space, Shift, Backspace, Enter).

Method

Twenty participants volunteered for this study (9 female; 11 male). They ranged in age from 19-48 ($M = 24.7$, $SD = 9.0$). Six participants had technical backgrounds, while the rest spanned domains such as social science, music, and administration. All participants reported having used a touch screen device for at least 6 months beforehand. Only one participant used an alternative to the standard QWERTY keyboard (Swype) to enter text on his device.

The study was conducted on an Apple iPad using a dynamic webpage written in JavaScript and HTML5. The webpage displayed full-screen in landscape orientation and the symbols and commands (26 in all) were presented in random order. Instructions were presented at the top of the screen, along with the buttons: “Done”, “Clear” and “Skip” (if applicable); the rest of the screen was blank for drawing. For each symbol or command, participants completed three gesture trials. To compare gesture similarity with and without a visual example for the symbols, Trial 1 only gave the symbol’s name (e.g., “comma”), while Trial 2 additionally provided four font-varied examples (e.g., ‘\$ \$ \$ \$’); for commands, Trials 1 and 2 were the same. After Trial 2, 7-point Likert scales were used to rate how easy the gesture was to draw and whether it was “a good match for its intended purpose” (following [8,9]). Finally, in Trial 3, participants were asked to switch how they drew the gesture: using multi-touch if they had previously used a single touch, or vice versa. Participants could opt to skip Trial 3 if they could not think of a reasonable gesture (e.g., multi-touch for ‘.’). The webpage logged all interactions.

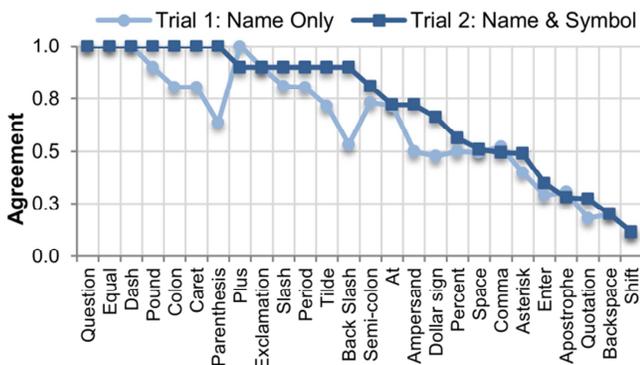


Figure 2. Agreement scores for each punctuation symbol or command: a score of 1.0 indicates perfect agreement, while 0.0 indicates that no two gestures agreed.

For each symbol and command, three members of our team independently grouped similar gestures (similar to [8,9]). Because certain symbols (e.g., ‘=’) intuitively afford multiple strokes that could be applied with a *single* multi-touch gesture or *multiple* single-touch gestures, our groupings focused on the shape of the stroke rather than the number of fingers used. After groupings were completed, they were combined into a final set using a majority voting process. Agreement was calculated slightly differently from [8,9] as the count of all pairs of gestures that were deemed similar, divided by the total number of possible pairs:

$$Agreement_s = \frac{\sum_{g \in G_s} \binom{|g|}{2}}{\binom{N}{2}} \quad (1)$$

where G is the set of groups for the symbol or command s and N is the total number of gestures drawn for s . Since N is 20 (the number of participants) in our case, the denominator is equal to 380, the total number of pairs of gestures (i.e., “ N choose 2”). Agreement ranges from 0 to 1, inclusive.

Results

Figure 2 shows agreement scores; higher scores indicate more guessable gestures. Average agreement across all symbols and commands was .63 ($SD = .27$) in Trial 1 and .71 ($SD = .29$) in Trial 2, which was a significant difference ($t_{25} = 3.807$, $p < .001$). Examining Trial 1 gestures in more detail suggests that symbol names were sometimes misinterpreted (e.g., 5 participants drew a forward slash instead of a backslash).

Half the punctuation symbols and commands had agreement scores of .90 or higher in Trial 2, suggesting that selecting the most frequently drawn gesture for those cases will allow for a highly guessable set of gestures. The four commands (Shift, Enter, Space, Backspace) resulted in significantly lower agreement scores than the symbols, even in Trial 1 before participants were given visual examples ($t_{24} = 4.009$, $p = .001$). Shift, for example, had an agreement score of only .12; Figure 3 shows the diversity of gestures.



Figure 3. Some examples of the 13 different *Shift* gestures, the right-most two of which are multi-touch gestures.

To assess whether single-touch or multi-touch gestures were preferred for each symbol or command, we calculated the maximum number of fingers concurrently on the screen at any point during the gesture. In Trial 1, multi-touch gestures were made by more than one participant only for the following: ‘Shift # “ : =’ (Trial 2 results were similar.) For those symbols and the Shift command, at most half of the participants used a multi-touch gesture, from 15% of participants for Shift to 50% for ‘=’. Interestingly, when participants were asked whether they had preferred single-touch or multi-touch for the four punctuation symbols just listed (‘# “ : =’), several participants who had *initially* used single-touch stated they ultimately preferred multi-touch: preference was 65%-70% for multi-touch for all four of

those symbols. Following these preferences, participants were more likely to skip Trial 3 for punctuation symbols that clearly had no appropriate multi-touch gesture (e.g., 17 out of 20 participants did so for ‘.’).

Subjective measures reflected the agreement scores (scales are 1-7; 7 = “strongly agree”). On average, gestures were rated high on ease, with punctuation symbols ($M = 6.2$, $SD = 0.5$) and commands ($M = 6.0$, $SD = 0.4$) not significantly different. However, the drawn gestures were considered to be a significantly better match for symbols ($M=6.5$, $SD=0.2$) than for commands ($M=5.4$, $SD=0.5$) ($t_{3,16}=7.827$, $p = .022$; df adjusted to account for unequal variance).

On average in Trials 1 and 2, participants took 1.2 seconds ($SD = 5.6$) to draw each gesture. These times ranged in Trial 2 from 0.23 seconds ($SD = .33$) for ‘.’ to 2.20 seconds ($SD = 0.72$) for ‘@’. Notably, gestures for Space, while relatively high in agreement, were unreasonably slow for such a common action ($M=0.95$ s, $SD=0.47$). Most commonly, space was drawn as a horizontal bracket (‘]’).

User-Defined Gesture Set for Non-Alphanumeric Input

Figure 4 shows the user-defined gesture set based on these results. The set contains all punctuation symbols tested, except for comma and period. These two symbols are more frequent than some letters [10] and the comma gesture conflicted with the parenthesis and apostrophe gestures; as such, we left comma and period as keys on the primary keyboard. Since the commands generally resulted in lower agreement and subjective scores than did the punctuation symbols, we recommend commands also remain as keys on the primary keyboard. (Our system also includes a popup pie menu for commands; see Figure 1b and next section.)

For the remaining symbols, we selected the most popular gesture for each symbol in Trial 1. The only potential conflict in this set is between parenthesis and apostrophe, which differ only in height: apostrophes were 1.56 cm ($SD = 1.20$); parentheses were 5.49 cm ($SD = 1.87$). Whether or not this is an issue will depend on the particular gesture recognizer used. Our data suggests that both single-touch and multi-touch gestures should be allowed for ‘= # : “’.

&	,	,	*	*	@	@
\	^	:	\$	\$	=	/
!	!	()	%	%	%
+	#	?	“	”	-	;

Figure 4. The final set of gestures with examples taken from study data. All are single-touch gestures, except ‘# “ : =’, for which we recommend multi-touch also be supported. Comma and period are not included here and remain as keys on the primary keyboard.

AN AUGMENTED QWERTY KEYBOARD

We implemented and evaluated a ten-finger touch-screen QWERTY keyboard incorporating the gesture set.

The System

Figure 1 shows screenshots of the system, built in C# 4.0 for a Microsoft Surface computer. The QWERTY keyboard is a standard touch-screen keyboard implementation with the same layout and key size (.09" × 0.9") as the device’s native keyboard. Gesture recognition is done using a modified version of the \$N Multistroke Recognizer [1], which uses template matching to classify gestures based on a library of examples. To extend the recognizer for multi-touch gestures and to reduce ambiguity in gesture recognition, we restricted the candidate template set based on the number of *concurrent* fingers and limited the possible gesture angle to be relative to the keyboard orientation. Note that we maintained \$N’s original invariance to gesture size.

The user invokes the gesture mode by placing 4+ fingers down with either hand (to accommodate left- and right-handed users) and draws the gesture with the opposite hand; see Figure 1a. The keyboard changes color from blue to orange to indicate the user is in gesture mode. The gesture is classified when both hands lift from the screen, and, finally, the resulting symbol is entered into the text stream. The software also includes a popup pie menu to access modifier keys *in context*, although we did not evaluate this aspect of the system in this study. This menu could contain input for which guessable gestures are not as obvious; Shift, Backspace, Control and Alt are shown in Figure 1b. The user invokes the menu by tapping and holding their finger over a key for 400 ms, then swipes to select the option they want. Lifting the finger without swiping will cancel.

Preliminary Evaluation

We conducted a preliminary study to: (1) gather subjective feedback on the gestures versus a standard moded-keyboard alternative; (2) assess the performance feasibility of the new technique—that is, determine if the gestural interface was at least on par with the moded-keyboard, even for novice users without much gesture learning.

Method

Six participants (2 female; 4 male) volunteered for this study. They were on average 29.3 years old ($SD = 2.9$). Five were right-handed; 1 was left-handed. All participants had been using a touch-screen device for at least 1 year and 5/6 entered text using a standard touch screen QWERTY keyboard; 1 participant used a gestural text input technique (Swype). As such, all participants had previous experience with moded-keyboards for punctuation symbols.

Study sessions lasted 30 minutes. The moded-keyboard used the same interface as the gesture condition, except that instead of gestures, two “Symbols” keys were used to toggle the key labels between alphabetic characters and punctuation symbols. (These keys were located where Shift

keys are found on most keyboards.) Participants practiced entering the following 10 common symbols with each technique: ‘! ? @ \$ & ; ; “ ()’. After 8 practice attempts with each symbol on each of the interfaces (gestures vs. moded-keyboard), we asked participants to complete a more realistic task: entering pairs of words separated by one of the punctuation symbols (e.g., ‘toil? and’). In the case of parenthesis and double quote, two words and two punctuation symbols were used (e.g., ‘with (sister)’). The interfaces were counterbalanced and participants entered 18 of these word-pairs with each.

Findings

Although we were primarily interested in subjective responses in this evaluation because we are not seeking to replace, but rather to augment, moded designs, we first briefly examined performance. On average, participants typed at 19.7 WPM ($SD = 5.16$) with the gestures and 19.9 WPM ($SD = 4.4$) with the moded-keyboard. Uncorrected error rates [7] were also similar, at 6.3% ($SD = 4.8$) for the gestures and 6.3% ($SD = 7.7$) for the moded-keyboard. These differences were not statistically significant. The results demonstrate the feasibility of the gestural interface, even for these novice users compared to the *status quo*.

In general, reactions to the gestural input were positive. For example, when first introduced to the technique, P2 said: “This is great. It feels very seamless.” Two participants also commented that the gestures were “fun” (P1, P4). This positive sentiment translated to an overall preference for the gestural interface: 5/6 participants predicted they would prefer the gestures over a moded-keyboard if both options were available on a large touch-screen device (e.g., tablet). One reason for this preference was efficiency, for example:

“My opinion of gesture things is that there’s a slightly higher learning curve but they’re usually faster once the fingers know how to do them.” (P1)

The gestures were also perceived to require less visual attention than the moded-keyboard. Four participants commented to this effect, for example:

“[For the moded-keyboard] I always had to look down at the symbols to find where they were. With the second set [gestures] I felt I didn’t have to look down really at all. If I did look down it was just to figure out if I was on the home row.” (P2)

Both interfaces also received some negative comments. The awkwardness of having to mode-switch arose with two participants, once for the moded-keyboard and twice for the gestures. The mode-switch mechanism would likely become easier with practice, but particularly so for the gestural interface since it does not require much visual attention (just putting 4+ fingers down). Future work is needed to confirm this hypothesis.

Finally, participants commented on the general difficulty of entering punctuation symbols on their touch-screen devices, which further motivates this research. P5 commented:

“It’s one of the things that stops me from typing on the iPad because it’s such a pain to get the punctuation correct.”

URLs, ‘@’ and passwords with a lot of punctuation were cited as particularly frustrating on touchscreen keyboards.

DISCUSSION AND CONCLUSION

We have introduced a bimanual, multi-touch gestural approach for non-alphanumeric text input that is designed to *augment*, not replace, existing solutions. This technique allows for input with low visual attention and does not require users to move their hands out of typing position. To inform our design, we first created a user-defined gesture set containing single-touch and multi-touch gestures based on data from 20 participants. Although preliminary, an evaluation of the full technique suggests that users prefer the gestural input over a moded-keyboard interface. We envision this new technique will complement the moded-keyboard approach. As a side benefit, the alternate symbol keyboards can act as a “reference sheet” for novice gesture users. Future work includes a more thorough evaluation and extending the gestural interface to tablets and smaller devices. The gesture mode could be invoked for two-thumb typing, for example, by holding a single finger down.

Acknowledgements

The authors would like to thank Jon Froehlich. This work was supported in part by NSERC, Microsoft Research, and the National Science Foundation under grant IIS-0811063. Any opinions, findings, conclusions or recommendations expressed in this work are those of the authors and do not necessarily reflect those of any supporter.

REFERENCES

1. Anthony, L. and Wobbrock, J.O. 2010. A lightweight multistroke recognizer for user interface prototypes. *Proc. GI '10*, 245-252.
2. Brewster, S. A., and Hughes, M. 2009. Pressure-based text entry for mobile devices. *Proc. MobileHCI '09*, 73-76.
3. Isokoski, P.. 2004. Performance of menu-augmented soft keyboards. *Proc. CHI '04*, 423-430.
4. Kristensson, P. and Zhai, S. 2004. SHARK²: a large vocabulary shorthand writing system for pen-based computers. *Proc. UIST '04*, 43-52.
5. MacKenzie, I.S., Soukoreff, R.W. 2002. Text entry for mobile computing: models and methods, theory and practice. *Human-Computer Interaction* 17 (2), 147-198.
6. MacKenzie, I.S., Zhang, S.X. 1997. The immediate usability of Graffiti. *Proc. GI '97*, 129-137.
7. 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. *Proc. CHI '03*, 113-120.
8. Wobbrock, J.O., Aung, H.H., Rothrock, B. and Myers, B.A. (2005). Maximizing the guessability of symbolic input. *Extended Abstracts CHI '05*, 1869-1872.
9. Wobbrock, J.O., Morris, M.R., and Wilson, A.D. 2009. User-defined gestures for surface computing. *Proc. CHI '09*, 1083-1092.
10. Zhai, S., Hunter, M., and Smith, B.A. 2002. Performance optimization of virtual keyboards. *Human-Computer Interaction* 17 (3), 89-129.