Avestia Publishing
International Journal of Virtual Worlds and Human-Computer Interaction
Volume 1, Year 2013
Journal ISSN: 2368-6103
DOI: 10.11159/vwhci.2013.001

Eye on the Message: Reducing Attention Demand for Touch-based Text Entry

I. Scott MacKenzie, Steven J. Castellucci
Dept. of Electrical Engineering and Computer Science
York University
Toronto, Ontario, Canada, M3J 1P3
mack@cse.yorku.ca; stevenc@cse.yorku.ca

Abstract- *A touch-based text entry method was developed with the goal of reducing the attention demand on the user. The method, called H4touch, supports the entry of about 75 symbols and commands using only four soft keys. Each key is about 20× larger than the keys on a Qwerty soft keyboard on the same device. Symbols are entered as Huffman codes with most letters requiring just two or three taps. The codes for the letters and common commands are learned after a few hours of practice. At such time, the user can enter text while visually attending to the message, rather than the keyboard. Similar eye-on-the-message interaction is not possible with a Qwerty soft keyboard, since there are many keys and they are small. Entry speeds of 20+ wpm are possible for skilled users.*

Keywords: Text entry, mobile text entry, touch-based input, visual attention, reducing attention demand.

1. Introduction

Improving text entry methods is among the oldest and most active areas of research in human-computer interaction. Even today, with the proliferation of touch-based mobile devices, text entry remains a common and necessary task. Whether entering a URL, an e-mail address, a search query, or a message destined for a friend or colleague, the interaction frequently engages the device's text entry method to enhance the user experience.

In our observations of users interacting with touch-based devices, it is evident that text is most commonly entered using a Qwerty soft keyboard, perhaps enhanced with swipe detection [26], word completion, next key prediction [9], or auto correction [23]. Because the keys are numerous and small, the user must attend to the keyboard to aim at and touch the desired key. Even highly practiced users must attend to the keys. There is little opportunity for eyes-free entry, due to the number and size of the keys. As well, the user cannot exploit the kinesthetic sense, as with touch typing on a physical keyboard. Consequently, input requires a continuous switch in visual attention – between the keyboard and the message being composed.

In this paper, we present a touch-based text entry method that reduces the attention demand on the user. The method uses a soft keyboard. However, there are only four keys and they are big. After a few hours of practice, the user can fully attend to the message being composed, rather than the keyboard. Entry speeds exceeding 20 words per minute are possible, as demonstrated later. The method provides access to all the required graphic symbols, modes, and commands, without additional keys.

2. Attention in User Interfaces

There is a long history of studying attention as it bears on human performance in complex systems and

Date Received: 2013-09-10
Date Accepted: 2013-11-07
Date Published: 2013-11-18

environments [12]. One area of interest is divided attention, where the human undertakes multiple tasks simultaneously. So-called *multitasking* or *time-sharing* [24, chapt. 8] is particularly relevant in mobile contexts since users frequently engage in a secondary task while using their mobile phone. Inevitably, performance degrades in one task or the other, perhaps both. A particularly insidious example is texting while driving, where statistics suggest up to a 23× increase in the risk of a crash while texting [19]. The article just cited is an example of the *correlational method* of research [13, chapt. 4], where data are gathered and non-causal relationships are examined. There are many such studies on the texting-while-driving theme [e.g., 4, 17, 18], and the conclusions are consistent: Danger looms when drivers divide their attention between controlling their vehicle and entering information on a mobile device.

There also studies using the *experimental method* of research [13, chapt. 4], where an independent variable is manipulated while performance data are gather on a dependent variable. Conclusions of a cause-and-effect nature are possible. If there is an interest in, for example, "number of crashes" as a dependent variable, then there are clearly both practical and ethical barriers to using the experimental method in the research. There are examples of such research, but such studies tend to use in-lab driving simulators. The results are consistent with the correlational studies, although the research questions are often narrowly focused on performance measures such as driving precision, lane keeping, velocity variability, and so on [e.g., 10, 21, 22].

Aside from driving, there are other situations where users engage in secondary tasks while using a mobile device. A common secondary task in the mobile context is walking. Numerous studies have examined performance effects while users walk while performing a task on a mobile device [2, 3, 6, 8, 25].

Our interest in attention is more narrow. We are looking within the text entry task, rather than in the relationship between text entry and other tasks. Yes, even within the text entry task, the user is required to divide his or her attention. Figure 1 illustrates. The user is entering a message on a touch-based mobile phone using a Qwerty soft keyboard. As noted earlier, the keys are numerous and small so the user's eyes are directed at the keyboard. Yet, the goal is creating a message, not tapping keys. So, the user is required to periodically shift attention to the message, to verify the correctness of the input.

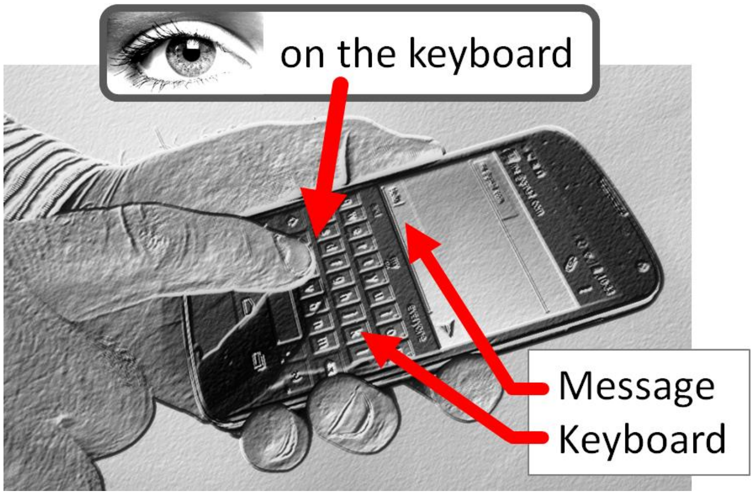


Figure 1. Text entry using a Qwerty soft keyboard: The eye must attend to the keyboard.

In fact, the situation is even worse. Most contemporary text entry methods also include word completion. To benefit from word completion, the user must examine a word list as entry proceeds. The word list typically appears just above the keyboard. Thus, the user's attention is divided among three locations: the keyboard, the message, and the word list.

Analyses within the text entry task commonly consider the *focus of attention* (*FOA*). One distinction for FOA is between a *text creation task* and a *text copy task* [16]. In a text creation task, the text originates in the user's mind, so there is no need to visually attend to the source text. If input uses a regular keyboard on a desktop computer and the user possesses touch-typing skill, the user need only focus on the typed text. This is a one FOA task. However, sometimes the user is copying text which is located in a separate space from where input occurs. This could be a sheet of paper beside a keyboard or text on a display. In this case the user must focus on both the source text and the entered text. This is a two FOA task. If input involves a mobile device and a soft keyboard, the situation is different still. A soft keyboard cannot be operated without looking at the keys. The user must look at the keyboard, the source text, and the entered text. This is a three FOA task. Of course, the user will divide his or her attention in whatever manner yields reasonable and efficient input. With word completion or prediction added, the task becomes four FOA. With driving or another secondary task added, FOA is further increased. Clearly, reducing the FOA demand on the user is important.

An interesting way to reduce the attention demand where word completion is used is seen in the Qwerty soft keyboard on the BlackBerry *Z10*, released in February 2013 (blackberry.com). To reduce the attention required to view the word list, predictions are dynamically positioned above the key bearing the next letter in the word stem. The same method is used in the *Octopus* soft keyboard (ok.k3a.me). For example, after entering BUS, several words appear above keys, including "business" above "I". See Figure 2. Instead of entering the next letter by tapping I, the user may touch I and flick upward to select "business". With practice, entry speeds have been demonstrated to exceed those for the Qwerty soft keyboard [5]. The important point here is that the user is attending to the I-key, so selecting "business" does not require shifting attention to a word list.

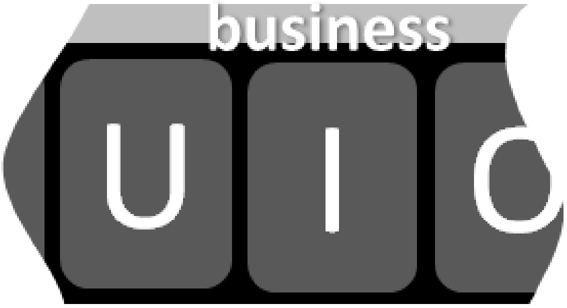


Figure 2. Partial view of BlackBerry *Z10* soft keyboard. See text for discussion.

While the approach in the BlackBerry *Z10* is notable, we are taking a different view to attention. Our goal is to enable the user to fully attend to the message being composed, much like the act of touch-typing on a physical keyboard. Before describing our method, a descriptive model is presented that relates the precision in an interaction task to the demand in visual attention.

3. Frame Model of Visual Attention

Visual attention is not simply a yes/no condition. The level of visual attention will vary according to the spatial precision demanded in the task. To express the idea of an attention continuum, we developed a descriptive model which we call the *frame model of visual attention* [14]. As a descriptive model, the goal is to support the interpretation and understanding of a problem space. Examples include Buxton's *three-state model for graphical input* [1] or Johansen's *quadrant model for groupware* [11]. Descriptive models are in contrast to predictive models [13, chapt. 7], wherein a mathematical expression is used to predict the outcome on a human performance variable based on task conditions. The frame model of visual attention is shown in Figure 3. Four levels are shown, but the model could be re-cast with different granularity depending on the interactions of interest. The intent is to show a progression in the amount of visual attention required for different classes of interaction tasks. High demand tasks are at one end, low demand tasks at the other. By "demand", we refer to the amount or precision in visual attention, not to the difficulty of the task.

The interactions in Figure 3 vary in the demand placed on the user to visually attend to the location where a human responder (e.g., a finger) interacts with a computer control (e.g., a touch-sensing display). Tasks in the *point frame* require a high level of precision, such as selecting a point or pixel. Consequently, a high level of visual attention is required. In the *target frame*, actions are slightly less precise, such as selecting an icon or a key in a soft keyboard. However, the task cannot be performed reliably without looking directly at the target; so there is still a demand on visual attention, although less so than in the point frame.

The *surface frame* applies to interactions that only require a general spatial sense of the frame or surface of a device. Broadly-placed taps, flicks, pinches, and most forms of gestural input apply. The visual demand is minimal: Peripheral vision is sufficient. The *environment frame*, at the right in Figure 3, includes the user's surroundings. Here, the visual frame of reference encompasses the user, the device, and the environment. In most cases, the demand for visual attention is low, and requires only peripheral vision, if any. Some interactions involving a device's accelerometer or camera apply. Virtual environments or gaming may also apply here, depending on the type of interaction.

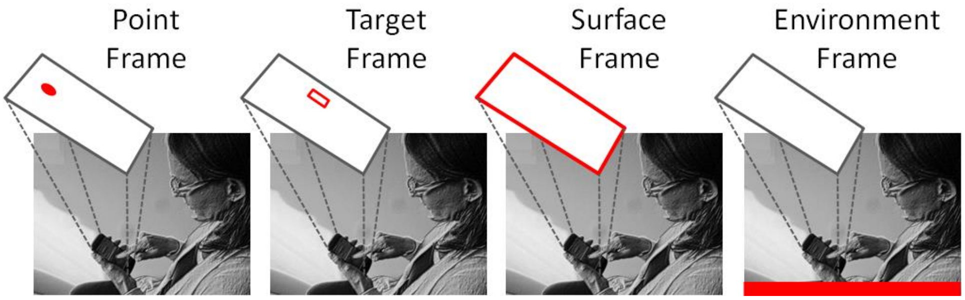


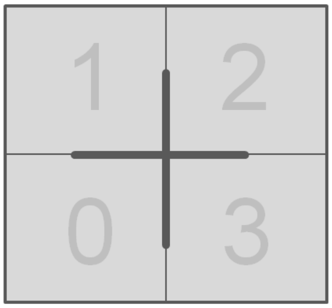
Figure 3. Frame model of visual attention. Less visual attention is required moving from left to right in the figure. See text for discussion.

As a descriptive model, the frame model of visual attention is a tool for thinking about interaction. The insight gained here is that touch-based text entry with a Qwerty soft keyboard operates in the target frame: Selecting keys requires the user to look at the keyboard. By relaxing the spatial precision in the task, interaction can migrate to the surface frame, thus freeing the user from the need to look at the keyboard. For this, we introduce *H4Touch*.

4. H4Touch

H4Touch was inspired by *H4Writer*, a text entry method using four physical keys or buttons [15]. The method uses base-4 Huffman coding to minimize the keystrokes per character (*KSPC*) for non-predictive English text entry. *KSPC* = 2.321, making it slightly less efficient than multitap on a standard phone keypad. Bear in mind that multitap requires nine keys, whereas *H4Writer* requires only four. This remarkably low *KSPC* comes by way of the base-4 Huffman algorithm: The top nine symbols (SPACE, e, t, a, o, i, n, s, h) each require only two keystrokes yet encompass 72% of English. A longitudinal evaluation of *H4Writer* with six participants yielded a mean entry speed of 20.4 wpm after ten sessions of about 40-minutes each [15]. The evaluation used thumb input with a 4-button game controller. *H4Touch* seeks to leverage the performance capability of *H4Writer* to the domain of touch input on mobile devices. The devices of interest are smart phones, pads, and tablets with touch-sensing displays.

Our touch implementation uses a square grid with keys and codes assigned as in Figure 4. About 75 symbols are supported, including letters, numerals, punctuation, UTF-8 symbols, ENTER, BACKSPACE, SHIFT, CAPSLOCK, etc. Symbols not shown in Figure 4 are in a separate code tree accessed via the SYM code (033).



SPACE	33	i	13	y	212
BKSPC	00	n	12	p	211
ENTER	01	s	31	g	210
SHIFT	02	h	10	b	3203
CAPLCK	030	r	322	v	3202
.	031	l	300	k	3201
,	032	d	321	x	32003
SYM	033	c	303	j	32002
e	11	u	302	q	32001
t	22	f	301	z	32000
a	23	m	323		
o	20	w	213		

Figure 4. *H4Touch* key codes for core symbols and commands.

A screen snap of the evaluation software running on an Android phone is shown in Figure 5a. With each tap, the symbol-to-key assignments are updated to show the progression through the Huffman code tree. Most letters require only two or three taps. The software is configured to present phrases selected at random from a set. Timing and keystroke data are collected beginning on the first tap, with performance data presented in a popup dialog after the last tap.

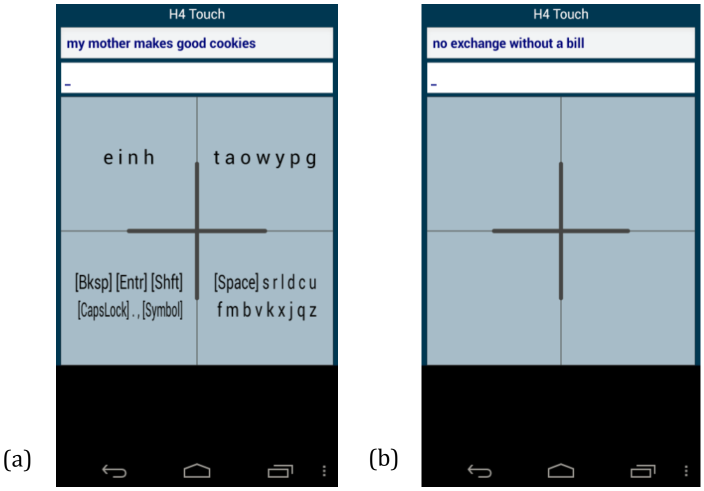


Figure 5. *H4Touch* screensnaps. (a) Symbol-to-key assignments are updated with each tap. (b) With experience, visual feedback is not needed.

Initial entry is visually guided, as users learn the codes for letters and common commands. Learning, then, takes place in the target frame, as depicted in Figure 3. Common letters and words are learned quickly. Entry speeds reach about 12-15 words per minute after a few hours of practice, as demonstrated in the initial evaluation of *H4Writer* [15]. At this point, the user has memorized the most common codes. And it is at this point that the most interesting aspect of *H4Touch* appears. Interaction migrates to the surface frame (see Figure 3) as the user now visually attends to the message, rather than to the keyboard. Of course, mistakes are made, but the feedback is immediate because the user is attending to the message. There is no need to switch attention between the keyboard and the message.

Not only are visual key assignments not needed for experienced users, they are potentially annoying. The preferred mode of interaction after a few hours of practice is seen in Figure 5b where the rendering of symbol-to-key assignments is suppressed. Of course, this is a choice to be made by the user. In the evaluation software, this option ("Hide letters") is enabled through a checkbox user preference. We should add that suppressing the output of symbol-to-key assignments only affects letters. Even experienced users are likely to need the key symbols when entering some of the more esoteric symbols or commands, which begin with SYM (033).

We offer Figure 6 for comparison with Figure 1 and to echo this paper's title. With *H4Touch*, the user's eye is on the message, not the keyboard. Of course, the caveat "after a few hours of practice" holds. More correctly, the caveat is "after a few hours of practice *and for all interaction thereafter*".

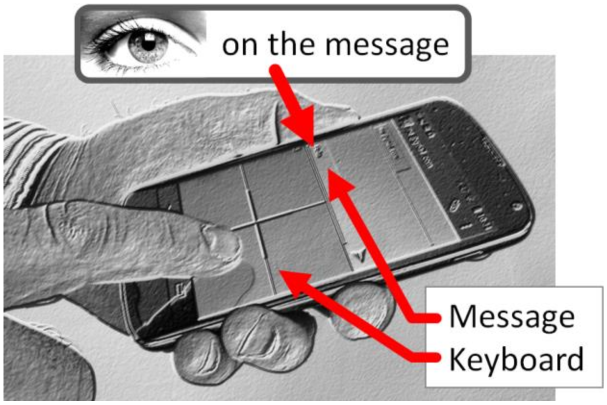


Figure 6. Text entry using *H4Touch*: The eye attends to the message.

No amount of practice with a Qwerty soft keyboard allows for a similar experience. There are simply too many keys and they are too small. By way of comparison, Figure 7 shows a single *H4Touch* key and a key from a Qwerty soft keyboard, with dimensions as measured on a Google *Nexus 4* in portrait mode. The *H4Touch* key is about 20× larger.

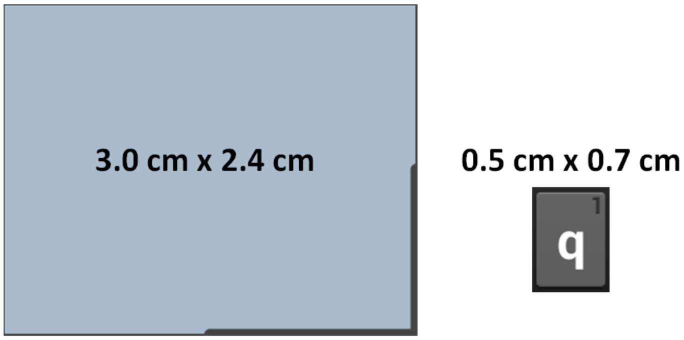


Figure 7. Key size comparison between *H4Touch* and a Qwerty soft keyboard. Dimensions measured on a Google *Nexus 4* in portrait mode.

5. User Experience

There are several techniques to improve the user experience with *H4Touch*. One technique is to use feedback to convey a sense of progress as text entry proceeds. Of course, the primary visual feedback is the appearance of text in the message being composed. As well, auditory and vibrotactile feedback options are selectable via checkboxes in a user preferences dialog. The feedback is provided on the last tap in a code sequence – after the second or third tap for most letters (see Figure 4). Intermediate feedback is provided by displaying the key code in red on the *H4Touch* keyboard. See Figure 8a.

Providing an option for two-thumb input in landscape mode is also important. See Figure 8b. There is the potential for faster entry with two thumbs, but testing for this with *H4Touch* has not been done.

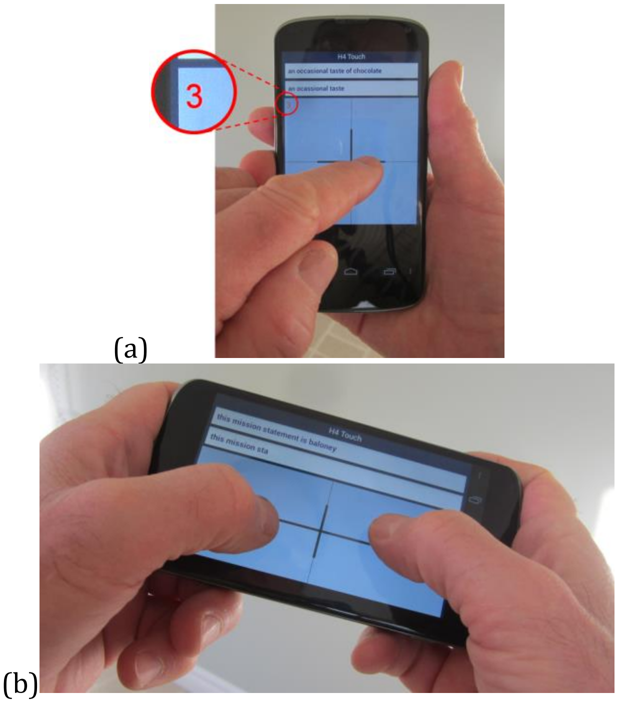

(a)
3
H4 Touch
an occasional taste of chocolate
an occasional taste
(b)
H4 Touch
this mission statement is baloney
this mission sta

Figure 8. *H4Touch* operation. (a) Portrait, showing Huffman code progress. (b) Landscape using two thumbs.

6. Extensibility

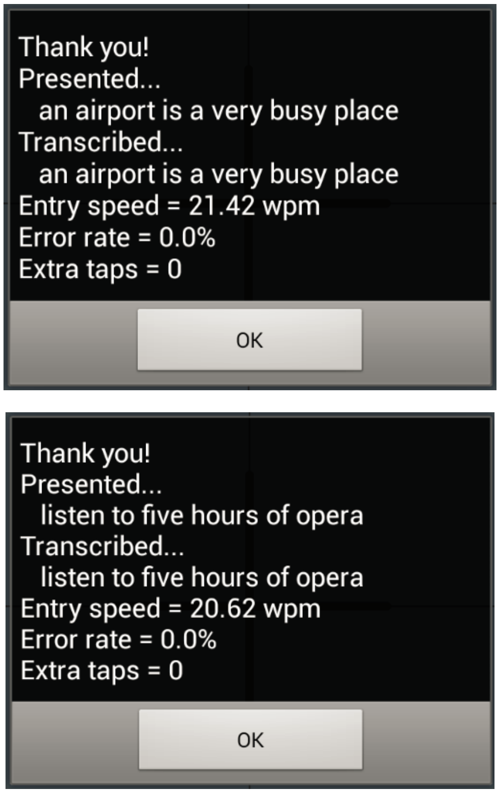
H4Touch currently supports about 75 symbols and commands. As noted in the original design of *H4Writer*, letter codes were generated using the base-4 Huffman algorithm along with letter probabilities for English [15]. The 0-branch of the code tree was hidden from the algorithm and manually designed later to include codes for commands and other symbols. The most common UTF-8 symbols are included in a "symbol tree" (SYM = 033). It is a simple process to extend the codes to include additional symbols, such as accented characters or other symbols from the broader set of Unicodes (e.g., UTF-16). For example, the symbol tree currently includes the dollar sign ("$" = 2200). To include symbols for other currencies, the codes may be adjusted as follows:

$ = 22000
€ = 22001
£ = 22002
¥ = 22003

Other design issues are pending, such as including code trees for other scripts, selected on an as-needed basis.

7. User Performance

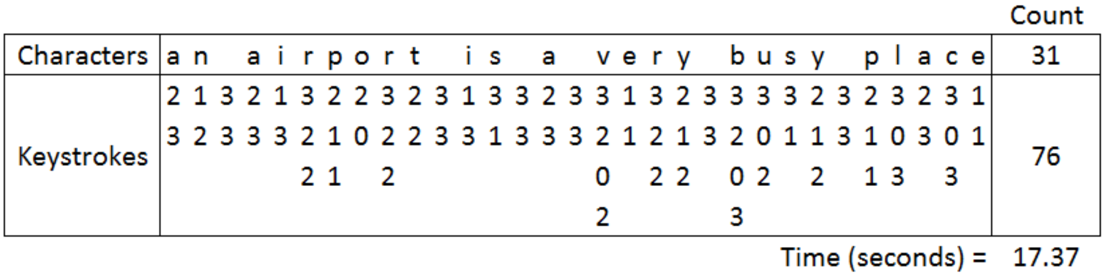
Preliminary testing with a user who has about 10 hours of practice suggests that entry speeds of 20 wpm are readily achievable. The codes are identical to those used with *H4Writer* and the interactions very similar: the index finger on 4 soft keys for *H4Touch* vs. the thumb on 4 physical buttons for *H4Writer*. Thus, the performance and learning trends demonstrated with *H4Writer* are expected to hold for *H4Touch*. Figure 9 provides examples of popup dialogs that appear after the input of sample phrases. These results are consistent with the initial evaluation of *H4Writer* [15]. The data collected for the left-side example in Figure 9 are illustrated in Figure 10.



```
Thank you!
Presented...
  an airport is a very busy place
Transcribed...
  an airport is a very busy place
Entry speed = 21.42 wpm
Error rate = 0.0%
Extra taps = 0
```

```
Thank you!
Presented...
  listen to five hours of opera
Transcribed...
  listen to five hours of opera
Entry speed = 20.62 wpm
Error rate = 0.0%
Extra taps = 0
```

Figure 9. Popup dialogs showing text entry performance.



																															Count
Characters	a	n		a	i	r	p	o	r	t		i	s		a		v	e	r	y		b	u	s	y		p	l	a	c	e	31
Keystrokes	2	1	3	2	1	3	2	2	3	2	3	1	3	3	2	3	3	1	3	2	3	3	3	3	2	3	2	3	2	3	1	76
	3	2	3	3	3	2	1	0	2	2	3	3	1	3	3	3	2	1	2	1	3	2	0	1	1	3	1	0	3	0	1	
						2	1		2								0		2	2		0	2		2		1	3		3		
																	2					3										

Time (seconds) = 17.37

Figure 10. Keystrokes, characters, and time for input of an example phrase using *H4Touch*. See also Figure 9.

The phrase included 31 characters which were entered in 17.37 seconds with 76 keystrokes (finger taps). The codes for each character are given in Figure 4. Relevant human performance statistics include entry speed, error rate, keystroke rate, and keystrokes per character (*KSPC*). The entry speed was (31 / 5) / (17.37 / 60) = 21.42 words per minute, as seen in the popup dialog in Figure 9.[1] The phrase was entered without errors, so the error rate was 0.0% (without any extra taps to correct errors). The keystroke rate was 76 / 17.37 = 4.4 keystrokes per second. This is fast but is well within the human capability for tapping, which peaks at about 6.5 taps per second.[2]

Keystrokes per character (*KSPC*) is both an *analytic metric* and an *empirical measure*. As an analytic metric, *KSPC* is defined as the number of keystrokes required, on average, for each character of text produced using a given input method using a given language. Used in this manner, *KSPC* is computed by analysis – without observing or measuring actual input from users. As noted in Section 4, *KSPC* = 2.321 for *H4Writer* [15].The same value holds for *H4Touch*, since the number of keys, the codes, and the language are the same. As an empirical measure, *KSPC* is calculated by observing interaction and gathering relevant data. The relevant data are the number of keystrokes entered and the number of characters produced. For the example in Figure 10, *KSPC* = 76 / 31 = 2.452. The analytic and empirical measures are slightly different, since the analytic measure is an overall average and the empirical measure is for a specific phrase of input. *KSPC* will rise if errors are made and corrected, but this was not observed for the phrases in Figure 9.

On a technical note, it is worth mentioning that initial tests of *H4Touch* on Android devices were disappointing. Although the evaluation software worked fine, the user experience suffered because the devices were unable to "keep up". Quick, light taps were occasionally missed, and this forced the user to slow down and to tap with slightly more, yet unnatural, force. This has changed with recent multi-core devices, such as the Google *Nexus 4*, which is quad-core and has a sensor sampling rate of 200 Hz. The device easily senses taps that are quick and light. The user experience with *H4Touch* is transparent: Performance is a human factor, not a technology factor.

8. Conclusion

We presented *H4Touch*, a 4-key touch-based text entry method for smart phones, tabs, and tablets. After a few hours of practice, the codes for the letters and common commands are memorized. Thereafter the user can enter text while visually attending to the message being composed, rather than the keyboard. A full evaluation was not undertaken, since *H4Touch* uses the same number of keys and the same codes as *H4Writer*. The latter method demonstrated performance of 20.4 wpm after ten sessions of practice. This is expected to hold for *H4Touch* and was demonstrated in testing with a single user.

A web site is setup to provide free access to the *H4Touch* software, including the source code file for the evaluation software (see Figures 5, 8, and 9) and the binary (apk) code for an input method editor (IME).[3] The IME is a soft keyboard that allows system-wide text entry in any application using *H4Touch*.

[1] It has been a convention dating to the 1920s to define "word" in "words per minute" as five characters, including letters, punctuation, etc. [7]

[2] The figure cited was measured experimentally by Soukoreff and MacKenzie [20] for continuous tapping on a single key. The apparatus was a stylus and soft keyboard.

[3] http://www.yorku.ca/mack/H4Touch/

References

[1] W. Buxton “A three-state model of graphical input”, Proceedings of the IFIP TC13 Third International Conference on Human-Computer Interaction - INTERACT '90, (Amsterdam: Elsevier, 1990), 449-456.

[2] T. Chen, S. Harper, and Y. Yesilada “How do people use their mobile phones? A field study of small device users” International Journal of Mobile Human-Computer Interaction, 3, 2011.

[3] J. Clawson, “On-the-go text entry: Evaluating and improving mobile text entry input on mini-Qwerty keyboards”, Doctoral Dissertation, Georgia Institute of Technology, 2012.

[4] J. L. Cook, and R. M., Jones, “Texting and accessing the web while driving: Traffic citations and crashes among young adult drivers”, Traffic Injury Prevention, 12, 2011, 545-549.

[5] J. Cuaresma, and I. S. MacKenzie, “A study of variations of Qwerty soft keyboards for mobile phones”, Proceedings of the International Conference on Multimedia and Human-Computer Interaction - MHCI 2013, (Ottawa, Canada: International ASET, Inc., 2013), 126.1-126.8.

[6] D. Fitton, I. S. MacKenzie, J. C. Read, and M. Horton, “Exploring tilt-based text input for mobile devices with teenagers”, Proceedings of the 27th International British Computer Society Human-Computer Interaction Conference – HCI 2013, (London: British Computer Society, 2013), 1-6.

[7] D. R., Gentner, “Keystroke timing in transcription typing”, in Cognitive aspects of skilled typing, (W. E. Cooper, Ed.). Springer-Verlag, 1983, 95-120.

[8] M. Goel, L. Findlater, and J. O. Wobbrock, “WalkType: Using accelerometer data to accomodate situational impairments in mobile touch screen text entry”, Proceedings of the the ACM SIGCHI Conference on Human Factors in Computing Systems - CHI 2012, (New York: ACM, 2012), 2687-2696.

[9] A. Gunawardana, T. Paek, and C. Meek, “Usability guided key-target resizing for soft keyboards”, Proceedings of Intelligent User Interfaces - IUI 2011, (New York: ACM, 2011), 111-118.

[10] S. T. Iqbal, Y.-C. Ju, and E. Horvitz, “Cars, calls, and cognition: Investigating driving and divided attention”, Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems - CHI 2010, (New York: ACM, 2010), 1281-1290.

[11] R. Johansen, “Groupware: Future directions and wild cards”, Journal of Organizational Computing and Electronic Commerce, 1, 1991, 219-227.

[12] S. W. Keele, Attention and human performance. Pacific Palisades, CA: Goodyear Publishing Company, Inc., 1973.

[13] I. S. MacKenzie, Human-computer interaction: An empirical research perspective. Waltham, MA: Morgan Kaufmann, 2013.

[14] I. S. MacKenzie, and S. J. Castellucci, “Reducing visual demand for gestural text input in touchscreen devices”, Extended Abstracts of the ACM Conference on Human Factors in Computing Systems - CHI 2012, (New York: ACM, 2012), 2585-2590.

[15] I. S. MacKenzie, R. W. Soukoreff, and J. Helga, “1 thumb, 4 buttons, 20 wpm: Design and evaluation of H4Writer”, Proceedings of the ACM Symposium on User Interface Software and Technology - UIST 2011, (New York: ACM, 2011), 471-480.

[16] I. S. MacKenzie, and S. X. Zhang, “An empirical investigation of the novice experience with soft keyboards”, Behaviour & Information Technology, 20, 2001, 411-418.

[17] M. Madden, and A. Lenhart, “Teens and distracted driving: Texting, talking and other uses of the cell phone behind the wheel”, Washington, DC: Pew Research Center, 2009, Retrieved 29/10/13 from http://pewresearch.org/assets/pdf/teens-and-distracted-driving.pdf.

[18] J. M. Owens, S. B. McLaughlin, and J. Sudweeks, “Driver performance while text messaging using handeld and in-vehicle systems”, Accident Analysis & Prevention, 43, 2011, 939-947.

[19] M. Richtel, “In study, texting lifts crash risk by large margin”, in The New York Times, 2009, July 28, Retrived 28/10/13 from http://www.nytimes.com/2009/07/28/technology/28texting.html.

[20] R. W. Soukoreff, and I. S. MacKenzie, “Theoretical upper and lower bounds on typing speeds using a stylus and soft keyboard”, Behaviour & Information Technology, 14, 1995, 370-379.

[21] R. J. Teather, and I. S. MacKenzie, “Effects of user distraction due to secondary calling and texting tasks”, Proceedings of the International Conference on Multimedia and Human-Computer Interaction - MHCI 2013, (Ottawa, Canada: International ASET, Inc, 2013), 115.1-115.8.

[22] J. Terken, H.-J. Visser, and A. Tokmakoff, “Effects of speed-based vs handheld e-mailing and texting on

driving performance and experience", Proceedings of the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications - AutomotiveUI 2011, (New York: ACM, 2011), 21-24.

[23] H. Tinwala, and I. S. MacKenzie, "Eyes-free text entry with error correction on touchscreen mobile devices", Proceedings of the 6th Nordic Conference on Human-Computer Interaction - NordiCHI 2010, (New York: ACM, 2010), 511-520.

[24] C. D. Wickens, Engineering psychology and human performance. New York: HarperCollins, 1987.

[25] K. Yatani, and K. N. Truong, "An evaluation of stylus-based text entry methods on handheld devices in stationary and mobile settings", Proceedings of the 9th International Conference on Human-Computer Interaction with Mobile Devices - MobileHCI 2007, (New York: ACM, 2007), 487-494.

[26] S. Zhai, and P.-O. Kristensson, "The word-gesture keyboard: Reimaging keyboard interaction", Communications of the ACM, (2012, September), 91-101.