

Gestural Text Entry Using Huffman Codes

Steven J. Castellucci, I. Scott MacKenzie

York University, Department of Computer Science and Engineering
 4700 Keele St., Toronto, Ontario, Canada M3J 1P3
 stevenc@cse.yorku.ca; mack@cse.yorku.ca

Abstract-The H4 technique facilitates text entry with key sequences created using Huffman coding. This study evaluates the use of touch and motion-sensing gestures for H4 input. Touch input yielded better entry speeds (6.6 wpm, versus 5.3 wpm with motion-sensing) and more favourable participant feedback. Accuracy metrics did not differ significantly between the two conditions. Changes to the H4 technique are proposed and the associated benefits and drawbacks are presented.

Keywords: Text entry, Huffman, H4, touch, motion-sensing

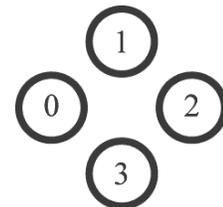
1. Introduction

Physical keyboards are convenient for desktop text entry, but they occupy significant space. Conversely, onscreen keyboards are popular for mobile computing, but having 27 or more keys in a small area imposes precise selection tasks on the user. Handheld devices usually have digitizers and gyroscopes, which can sense coarse gestures. Our paper investigates using such inputs to facilitate robust text entry.

Huffman codes (Huffman 1952) can be generated for text entry characters using the character frequency distribution of a corpus. The codes have two valuable properties: (1) no code forms a prefix to another code; and (2) encoded messages are of minimum average length. H4-Writer (MacKenzie et al. 2011) (abbreviated in this paper as H4) generates Huffman codes from four symbols ('0', '1', '2', and '3') and maps these symbols to four keys. Encodings for 27 characters are listed in Table 1. H4 allows for one-handed (one thumb) entry of letters, digits, punctuation, and symbols. With practice, input can be done eyes-free, and using only four keys greatly reduces selection complexity. Furthermore, the prefix-free property of Huffman codes means that input can be continuous, without the need for a pause or finger-up event to segment character input. In addition, the property (2) of Huffman codes means that H4's 2.3 keystrokes per character (KSPC (MacKenzie 2002)) is minimal.

Table 1. H4-Writer encodings for English. Encodings correspond to the gamepad buttons shown.

Space 33	e 11	t 22	a 23	o 20	i 13	n 12	s 31	h 10	r 322
l 300	d 321	c 303	u 302	f 301	m 323	w 213	y 212	p 211	g 210
b 3203	v 3202	k 3201	x 32003	j 32002	q 32001	z 32000			



Our over-arching goal is to implement an H4 keyboard for mobile devices. However, with mobile computing, physical keys and buttons have given way to touchscreens, accelerometers, and gyroscopes. This is aptly depicted by the plethora of smart phones and tablets commercially available. To determine

the ideal input method for a mobile H4 keyboard, we evaluated text entry using touch and motion-sensing input. We hope that our results will also aid other researchers investigating mobile interaction techniques.

After summarizing related input methods, we detail our two input techniques. We then detail a user study to evaluate our techniques, present the results, and elaborate on the findings.

2. Related Work

Like H4, Minimal Device Independent Text Input Method (MDITIM) (Isokoski and Raisamo 2000), EdgeWrite (Wobbrock et al. 2003), and Left, Up, Right, Down Writer (LURD-Writer) (Felzer and Nordmann 2006) text entry methods also encode characters using four discrete inputs. With MDITIM, the encoding symbols represent the directions up, down, left, and right. The resulting encodings are prefix-free. Users enter text by performing inputs in the desired directions. For example keyboard input requires pressing the arrow keys, while mouse input requires moving it in the desired directions. However, because of its device-independent nature, the directional encodings remain consistent. With EdgeWrite, each character corresponds to a unique sequence of corner selections. The use of a physical boundary around the input area is the primary characteristic of EdgeWrite. It facilitates accurate input for both able-bodied and motor-impaired users (Wobbrock et al. 2003). However, EdgeWrite sequences are not prefix-free, and instead rely on an input event (e.g., finger-up) to segment character input. With both text entry methods, the gesture alphabet, and the encodings, were designed to resemble the corresponding character in the Roman alphabet (Isokoski and Raisamo 2000, Wobbrock et al. 2003). LURD-Writer (Felzer and Nordmann 2006) uses mouse movement to select one of four keys on an onscreen keyboard. Once a key is selected, the user presses the left mouse button to activate the key, or presses the right mouse button to activate the key with the shift modifier enabled (i.e., for the uppercase letter or the associated symbol). None of the three methods have encodings based on Huffman's algorithm and, thus, are not of minimum average length.

Touch gestures have been used as input in previous text entry methods. Like H4, the QuikWriting (Perlin 1998) and Cirrin (Mankoff and Abowd 1998) techniques allow users to draw a continuous path to enter text of unlimited length. Characters are entered based on where the path enters and exits specific regions. However, those techniques divide the input area into 9 and 27 relatively small regions, respectively. This makes eyes-free input difficult, if not impossible. The ShapeWriter technique (Zhai and Kristensson 2003) provides an alternative to tapping on an onscreen keyboard. It associates each dictionary word with a path (i.e., shape) overlaid onto the keyboard. The path starts at the first letter of the word, intersects each subsequent letter, and ends at the last one. However, the finger-up event that segments each word prevents continuous text entry. Furthermore, depending on the layout of the keyboard, some words might have similar paths. When this occurs, the user selects the desired word from a menu of likely candidates (Zhai and Kristensson 2003).

Some existing text entry methods use mid-air gestures for input. UniGest (Castellucci and MacKenzie 2008) encodes characters using a pair of linear and rotational gestures performed mid-air. However, the encoding is not prefix-free, preventing continuous input. The TiltText (Wigdor and Balakrishnan 2003) technique uses the tilt of a mobile phone to disambiguate character input from a standard 12-key phone keypad. Hex (Williamson and Murray-Smith 2005) uses device orientation to navigate onscreen keyboards. Characters are arranged in six groups, displayed as hexagons. Tilting the device selects a group and redistributes the group's characters to the hexagons. Tilting the device again inputs a character and returns the hexagons to the initial layout. Unfortunately, this encoding limits the number of supported characters to 36. While this is sufficient for entering the 26 letters of the English alphabet plus 10 other symbols, H4 has no upper limit and already provides input for 26 letters and over 23 symbols (MacKenzie et al. 2011).

3. Input Techniques

We propose two new interaction techniques for H4 text entry:

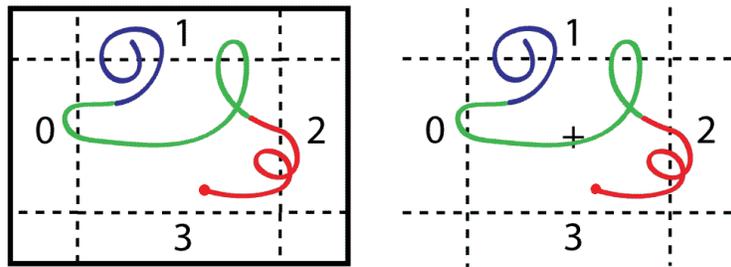


Fig. 1. The regions used for symbol input in the touch (left) and motion-sensing (right) techniques. In both images, the path (starting at the dot) represents the input sequence ‘t’ (red), ‘h’ (green), and ‘e’ (blue).

3. 1. Touch

Originally, H4 mapped each base-4 digit to a gamepad button. In our touch technique, each digit is mapped to a region along the outer edge of a touch-sensitive device in absolute pointing mode. The centre and corner regions are left unassigned (Fig. 1, left). When one drags a finger into a region, the corresponding digit is inputted. Continuous input can be accomplished by dragging a finger from one region to another. However, inputting the same digit consecutively would require re-entering that region from the centre of the touchpad. A raised edge around the touch-sensitive device allows the user to perform input without visual attention to the input area.

3. 2. Motion-Sensing

With motion-sensing input, we associate left, up, right, and down motions with the H4 encoding symbols. The motions are relative to a rest position or “origin” so that a continuous sequence of inputs can be made without an uncomfortable amount of displacement. This arrangement is analogous to touch input regions in mid-air, where the origin would be the centre of the touchpad (Fig. 1, right). In the Touch condition, symbol input (e.g., “1”) involved moving (dragging) one’s finger upwards into the “1” region. In the Motion condition, the same input also involved moving (tilting) the device upwards into the “1” region. We believe that this similarity in input mapping allows for evaluation the techniques, even though they use different muscle groups. In comparison, the EdgeWrite technique has been evaluated using various techniques that also use different muscle groups (Wobbrock and Myers 2005).

4. Method

4. 1. Participants

Eight paid participants (six males, two females) were recruited from our department. Ages ranged from 24 to 32 years ($\mu = 28$; $\sigma = 2.77$). All participants were familiar with using touch and motion-sensitive devices. All participants were also familiar with the H4 technique, having participated in a previous H4 study. We believed that using experienced participants would minimize H4 learning effects and yield results characteristic of the input methods. Though requiring H4 experience resulted in fewer participants than we had hoped, other published text entry studies have still produced significant finding using only five to nine participants (Jones et al. 2010, Koltringer et al. 2007, Li et al. 2011, Urbina and Huckauf 2010).

4. 2. Apparatus

We did not have a programmable smart phone available at the time of the study. To provide a common platform for each technique, we used a series R51 *ThinkPad* laptop (model 1836), running *Windows XP* and *Java* 1.6. The laptop’s built-in touchpad was used for touch input. Its dimensions

were 61 mm by 41 mm, with a 3 mm ridge along its edge. The touchpad, which usually emulates relative mouse input, was set to absolute mode for this experiment. To setup and manage experimental sessions, the onscreen mouse pointer was instead controlled using the laptop’s built-in isometric joystick. For motion-sensing input, we used a Nintendo *Wii Remote (Wiimote)* with the *Wii MotionPlus* gyroscope accessory. An MSI *Star Key* Bluetooth adapter facilitated communication between the *Wiimote* and the laptop. A gamepad warm-up session was used to allow comparisons between this study and the original H4 study (MacKenzie et al. 2011). For gamepad input, we used a Logitech *Dual Action* (Fig. 2).



Fig. 2. The gamepad (left) used for warm-up sessions, and the touchpad (center) and *Wiimote* (right) used in the two experimental conditions.

Fig. 3 illustrates the Java program used for the text entry task and for gathering performance and accuracy data. Although H4 text entry can be eyes-free, we displayed an onscreen keyboard. This approach eliminated any long delays associated with participants pausing to recall long or infrequent symbol sequences, thus increasing the number of phrases administered.

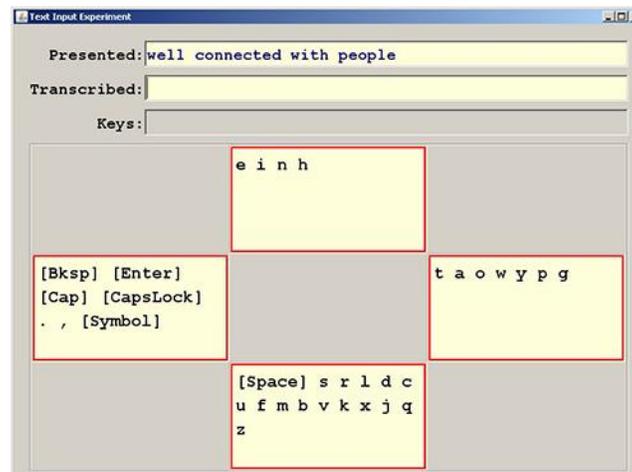


Fig. 3. The interface used to administer the text entry task.

Using the *GlovePIE* framework (<http://glovepie.org>), we wrote scripts to activate the H4 “keys” (outlined in red). Input events, such as gamepad button presses, specific *Wiimote* movement, and absolute position input from the touchpad, were converted to presses of numeric keys 0-3. Each numeric key represented the input of a symbol from the Huffman encoding alphabet and activated the corresponding H4 key.

In the default arrangement (as depicted in Fig. 3), characters are assigned to the key that represents the first encoding symbol. Once a key is activated, characters are assigned to the key that represents the second encoding symbol. All non-activated characters are removed from the arrangement. This reassignment continues until a key with only one character is activated, thus completing a Huffman code. The corresponding character is then entered and the character arrangement returns to the default one. Different audio cues were used to indicate key activation, entry of a letter, and input of “[Enter]”.

4. 3. Procedure

Participants began by entering fifteen phrases using the gamepad. This reproduced their previous H4 experience and provided a baseline comparison for the other input devices. Participants were instructed to “enter text as quickly as possible”. Participants were also told to correct errors, but to ignore errors they made two or more characters back.

The touch condition mapped input to four regions of the touchpad (as in Fig. 1, left). When one’s finger moved into the left, top, right, or bottom region, the respective H4 key was activated. The height and width of the top and bottom regions spanned 20% and 60% of the touchpad, respectively. The left and right regions used the opposite dimensions.

The motion-sensing condition mapped input to four *Wiimote* motions: left, up, right, and down (as in Fig. 1, right). Participants would hold the B-button (on the underside of the *Wiimote*) during input to distinguish an input gesture from other, general movement. Upon pressing the B-button, the *Wiimote*’s current orientation was set as the “origin”. Movement of 30 degrees from the origin (20 degrees for down) activated the corresponding H4 key. The decision to reduce the threshold for a down gesture was the result of wrist discomfort during a pilot study and research (Jones et al. 2010) showing reduced wrist movement in downward gestures.

Before each condition, participants were instructed on how to use the corresponding device. Then, participants entered a random practice phrase. At the conclusion of the user study appointment, the participant completed a questionnaire to gather device feedback and demographic information. Study appointments typically lasted one hour and took place in a quiet office, with participants seated at a desk.

4. 4. Design

The experiment employed a within-subjects factor, technique, with two levels: touch and motion-sensing. As previously mentioned, a warm-up session using a gamepad was used to allow comparisons between this study and the original H4 study (MacKenzie et al. 2011); it was not considered an experimental condition. The choice not to evaluate H4 eyes-free was made to increase the number of phrases per session. Each technique was used to enter fifteen phrases, each terminated with “[Enter]”. Phrases were chosen randomly (without replacement) from a 500-phrase set (MacKenzie and Soukoreff 2003). The phrases were converted to lowercase letters and did not contain any numbers or punctuation.

The dependent variables were entry speed and accuracy. For each phrase, timing for entry speed started with the first H4 key activation. This allowed the participant to take a break as needed between phrases. Timing stopped with the input of the final transcribed character; the time to input “[Enter]” was not included. Entry speed was calculated by dividing the length of the transcribed text by the entry time (in seconds), multiplying by sixty (i.e., seconds in a minute), and dividing by five (the accepted word length (Yamada 1980 p. 182)). The entry speed was averaged over the fifteen phrases and reported in words-per-minute (wpm).

Accuracy was measured according to the total error rate (TER), corrected error rate (CER), and uncorrected error rate (UER) metrics (Soukoreff and MacKenzie 2004). TER is the sum of CER and UER. UER uses the minimum string distance (MSD) metric (Soukoreff and MacKenzie 2001) to measure how different the transcribed text is from the presented phrase. In contrast, CER is the ratio of “[Bksp]” inputs to all character inputs. Error rates were averaged over the fifteen phrases and reported as a percent.

5. Results and Discussion

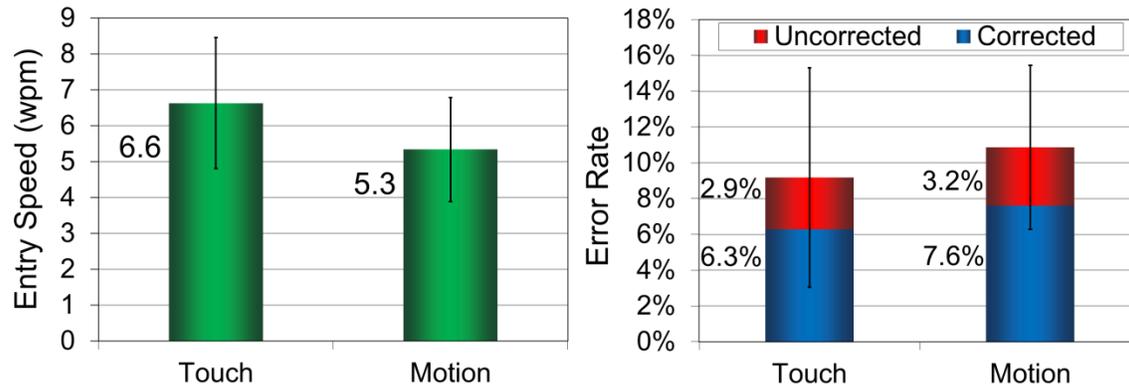


Fig. 4. The entry speeds (left) and error rates (right) for the two experimental conditions. Error bars show ± 1 standard deviation.

5. 1. Entry Speed

On average, participants achieved entry speeds of 6.6 wpm in the touch condition and 5.3 wpm in the motion-sensing condition. In comparison, these values represent 77% and 66% of the gamepad session's 9.7 wpm. Because we employed participants skilled in H4 input, the decreases in performance can be attributed to the interaction techniques themselves, rather than learning the H4 encodings. The performance results of our study indicate using touch input for mobile H4 text entry is preferable to motion-sensing in term of performance. Analysis of Variance (ANOVA) showed that technique had a significant effect on entry speed ($F_{1,6} = 8.95, p < .05$), with touch input faster than motion-sensing input. In addition, counterbalancing proved effective, as there was no group effect ($F_{1,6} = 0.10, ns$). Entry speed results appear in Fig. 4, left.

Gamepad entry speed in our study is greater than the Session 1 speed of 7.7 wpm in the original, longitudinal H4 study, but it is less than half of the 20.4 wpm reached by Session 10 (representing about 400 minutes of practice) (MacKenzie et al. 2011). Although participants were familiar with H4, a lack of practice significantly deteriorated their proficiency with the input technique. Additional training would refine the motor skills (i.e., “muscle memory”) specific to each input device and thus improve entry speed. However, considering that the directional mapping of H4 symbols is identical for both methods (Fig. 1), the relative performance difference between touch and motion-sensing input would likely remain the same.

In comparison, a longitudinal evaluation of MDITIM reported speeds of 2-8 wpm over ten 30-minute sessions (Isokoski and Raisamo 2000). In addition, the entry speed for our touch condition is identical to the 6.6 wpm reported for EdgeWrite (Wobbrock et al. 2003). An author of the Hex paper reported typing 10-12 wpm using that text entry method. However, that speed was reached after about 30 hours of training (Williamson and Murray-Smith 2005).

5. 2. Accuracy

The total error rate (TER) values for the touch and motion-sensing conditions were 9.2% and 10.9%, respectively. Corrected (CER) and uncorrected error rates (UER) appear in Fig. 4, right. Interestingly, there was no significant effect of technique on TER ($F_{1,6} = 0.70, ns$), CER ($F_{1,6} = 0.80, ns$), or UER ($F_{1,6} = 0.08, ns$).

The TER value for the gamepad session was 6.1% – lower than both the touch and motion-sensing conditions. The original H4 study also used a gamepad. It assessed accuracy using UER and reported an error rate of 0.69%. While our gamepad session had a lower UER value of only 0.1%, our touch and motion-sensing conditions had higher values. The higher error rates for the touch and motion-sensing

conditions could be attributed to the novelty of the interaction techniques. Because we used participants from previous H4 studies, they were familiar with using a gamepad for input.

Although many papers introducing text entry methods fail to mention any accuracy results, EdgeWrite reported a UER of 0.34% (Wobbrock et al. 2003). MDITIM showed a very high initial error rate of about 15%, but reported an “average error level over the whole experiment” of only 4.6% (Isokoski and Raisamo 2000). This value is described as “the percentage of written characters that were wrong”, but does not clarify whether it represents entered (but corrected) characters, or characters in the transcribed string.

5. 3. Participant Feedback

Participants favoured the touch condition in accuracy, required mental effort, and wrist comfort. This is illustrated in Fig. 5. We used the Mann-Whiney test for statistical significance in our two-sample, non-parametric participant feedback. Only wrist fatigue met the 5% threshold for significance ($U = 7.5, p < .05$). In general, five of the eight participants preferred touch input to the motion-sensitive input. Participants reported minor finger fatigue in the touch condition and considerable wrist fatigue in the motion-sensing condition.

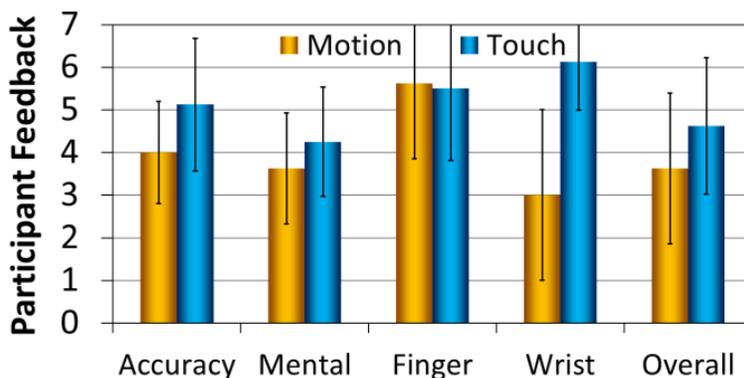


Fig. 5. Participant feedback scores. Error bars show ± 1 standard deviation.

Fatigue could be minimized by changing the mapping of encoding symbols to H4 keys. The rearrangement could take advantage of asymmetries in finger (thumb) and wrist movement (Jones et al. 2010) that make some motions less strenuous than others. Symbol rearrangement would also benefit from an analysis of gesture accuracy. For example, did the user move in a top-left or bottom-left direction when only a left motion was required? By determining which gestures (if any) were particularly difficult to execute accurately, frequent symbols could be mapped to more reliable gestures. Rearrangement would still preserve the beneficial Huffman code properties. However, skilled H4 users would need to unlearn the existing mappings (to avoid confusion) before learning the new ones. Alternatively, the shape and/or dimensions of an input region could be modified to accommodate user tendencies. Further investigation is required to determine the net benefit of any change to the technique.

6. Conclusion

We compared two methods of H4 text entry: touch input and motion-sensing input. Entry speed was significantly faster with touch input. Accuracy was slightly better with touch input, but not significantly so. Participants also preferred touch input over motion-sensing, though the techniques caused finger and wrist fatigue, respectively.

Given the choice of touch or motion-sensing input for a mobile keyboard, our study shows entry speed and user preference strongly favour using touch input for text entry. Further development of

the H4 technique could involve rearranging the H4 keys. If so, the data gathered from this study could be used to determine an arrangement that is beneficial for speed, accuracy, and user comfort.

References

- Castellucci, S.J. and MacKenzie, I.S. (2008). UniGest: Text entry using three degrees of motion "Ext. Abs. CHI 2008," Florence, Italy, pp. 3549-3554.
- Felzer, T. and Nordmann, R. (2006). Alternative text entry using different input methods "Proc. ASSETS 2006," Portland, Oregon, pp. 10-17.
- Huffman, D.A. (1952). A Method for the Construction of Minimum-Redundancy Codes. Proc. IRE 40, 1098-1101.
- Isokoski, P. and Raisamo, R. (2000). Device independent text input: A rationale and an example "Proc. AVI 2000," Palermo, Italy, pp. 76-83.
- Jones, E., Alexander, J., Andreou, A., Irani, P. and Subramanian, S. (2010). GesText: Accelerometer-based gestural text-entry systems "Proc. CHI 2010," Atlanta, Georgia, pp. 2173-2182.
- Koltringer, T., Isokoski, P. and Grechenig, T. (2007). TwoStick: writing with a game controller "Proc. GI 2007," Montreal, Quebec, pp. 103-110.
- Li, F.C.Y., Guy, R.T., Yatani, K. and Truong, K.N. (2011). The 1line keyboard: A QWERTY layout in a single line "Proc. UIST 2011," Santa Barbara, California, pp. 461-470.
- MacKenzie, I.S. (2002). KSPC (keystrokes per character) as a characteristic of text entry techniques "Proc. MobileHCI 2002," Pisa, Italy, pp. 195-210.
- MacKenzie, I.S. and Soukoreff, R.W. (2003). Phrase sets for evaluating text entry techniques "Ext. Abs. CHI 2003," Ft. Lauderdale, Florida, pp. 754-755.
- MacKenzie, I.S., Soukoreff, R.W. and Helga, J. (2011). 1 thumb, 4 buttons, 20 words per minute: Design and evaluation of H4-Writer "Proc. UIST 2011," Santa Barbara, California, pp.
- Mankoff, J. and Abowd, G.D. (1998). Cirrin: A word-level unistroke keyboard for pen input "Proc. UIST 1998," San Francisco, California, pp. 213-214.
- Perlin, K. (1998). Quikwriting: Continuous stylus-based text entry "Proc. UIST 1998," San Francisco, California, pp. 215-216.
- Soukoreff, R.W. and MacKenzie, I.S. (2001). Measuring errors in text entry tasks: An application of the Levenshtein string distance statistic "Ext. Abs. CHI 2001," Seattle, Washington, pp. 319-320.
- Soukoreff, R.W. and MacKenzie, I.S. (2004). Recent developments in text-entry error rate measurement "Ext. Abs. CHI 2004," Vienna, Austria, pp. 1425-1428.
- Urbina, M.H. and Huckauf, A. (2010). Alternatives to single character entry and dwell time selection on eye typing "Proc. ETRA 2010," Austin, Texas, pp. 315-322.
- Wigdor, D. and Balakrishnan, R. (2003). TiltText: Using tilt for text input to mobile phones "Proc. UIST 2003," Vancouver, Canada, pp. 81-90.
- Williamson, J. and Murray-Smith, R. (2005). Hex: Dynamics and probabilistic text entry "Proc. Switching and Learning in Feedback Systems," pp. 333-342.
- Wobbrock, J.O. and Myers, B.A. (2005). Gestural text entry on multiple devices "Proc. ACCESS 2005," Baltimore, Maryland, pp. 184-185.
- Wobbrock, J.O., Myers, B.A. and Kembel, J.A. (2003). EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion "Proc. UIST 2003," Vancouver, Canada, pp. 61-70.
- Yamada, H. (1980). A Historical Study of Typewriters and Typing Methods: from the Position of Planning Japanese Parallels. Journal of Information Processing, 2, 175-202.
- Zhai, S. and Kristensson, P.-O. (2003). Shorthand writing on stylus keyboard "Proc. CHI 2003," Ft. Lauderdale, Florida, pp. 97-104.