Comparing Two Methods for Gesture based Short Text Input using Chording.

Hendrik Witt
TZI, Wearable Computing Lab.
University of Bremen
D-28359 Bremen, Germany
hwitt@tzi.de

Torben Janssen
TZI, Wearable Computing Lab.
University of Bremen
D-28359 Bremen, Germany
torben@tzi.de

CHI 2007, April 28 – May 3, 2007, San Jose, USA
ACM 978-1-59593-642-4/07/0004.

Abstract

We report on the design and initial evaluation of two methods for short text input in wearable computing applications using hand gestures. A wireless data glove able to recognize 4 basic gestures is used together with the chording principle. We present two different concepts to map gestures to characters. A presentation of preliminary experiment results shows that simple free hand gestures in combination with different key maps are easier to learn and allow faster typing than distinct gestures assigned for each character.

Keywords

Text Input, Gesture, Chording, Wearable Computing

ACM Classification Keywords

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

Introduction

Gesture interaction has become a popular and important interaction technique not only for virtual reality (VR) or 3D user interfaces. Even recently launched consumer electronics, such as the Nintendo Wii system, feature gesture based game control by using inertial sensors integrated in the game controller.

Today, in wearable computing text input is typically done with the Twiddler [3,6] cording keyboard. Although the Twiddler has been shown appropriate to achieve high speed typing [6], it requires non negligible learning efforts. Because in some applications "hands-free" operation is required, the Twiddler is sometimes not appropriate as it needs to be carried around and held in hand while typing. In particular, wearable applications for industrial environments often call for short text or abbreviation input rather than full text input which can be also entered with input devices other than keyboards.

Glove based gesture interfaces were shown as alternatives to implement "hands-free" interaction [8]. Since it is difficult to recognize many complex gestures with high accuracy, the chording principle can be used instead of mapping gesture combinations to characters of the alphabet. Like an accord on the piano, gestures and keys can be presses in combination to encode more characters than the sum of gestures and buttons available on the input device would imply.

Related Work

Although gesture based text input has already been considered, much work focuses on the implementation of a "virtual keyboard". In [1,4] conductive materials detect finger pinches for text input. In [7], a chording glove for text input is shown. Instead of using free hand gestures, different buttons on the glove can be pressed to enter text. In [5], computer vision is used for gesture based text input. However, systems that use inertial sensors are rare. Scurry [2] uses a set of different inertial sensors and can be used to control a mouse or keyboard by moving the hand in the air.

The Data Glove

As shown in figure 1, a leather glove is used to mount the hardware components of our prototype glove. It features a 3D accelerometer and three buttons that can be used to recognize gestures. Only the buttons located on the fore- and middle finger were used by our text input methods. The third button, located on the ring finger, was omitted due to pre-tests that uncovered problems of subjects trying to reach and press it. The glove is connected via Bluetooth to the computer; offering a high degree of mobility.

The gesture recognition software used is able to recognize up to eight different hand gestures plus the state of the buttons. The supported gestures are: *left* and *right rotations*, *up* and *down* movements, as well as some of their combinations, e.g., a left rotation of the hand followed by moving the hand upwards. The software recognizes therefore up to 34 gestures; each gesture alone, as a chord together with one button, or as a chord where two buttons are pressed in a row.

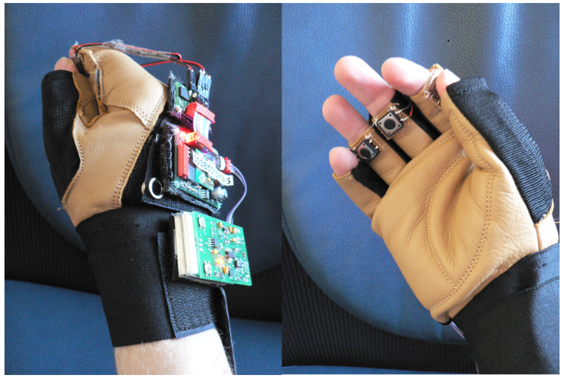

Figure 1: The data glove used for gesture based text input.

Design of the Text Input Methods

With 34 distinct gestures, an obvious approach for text input is to directly map a single gesture onto each of the 26 letters in the alphabet and the remaining 6 gesture to punctuation marks, space, caps lock, etc. Since some of the 34 available gestures are, however, complicated to perform without risking ambiguities that reduce gesture recognition accuracy and require higher learn effort, we deemed a reduction of gestures to be more beneficial in terms of learn ability and correct classifications that result in higher typing speed. To validate our hypothesis, one input methods was designed that utilizes only a subset of four easy to memorize hand gestures while the other method uses the entire set of gestures available.

Method 1

Besides performing only one of four distinct hand gesture or button presses at a time, gestures and buttons are combined to chords with either one or two buttons. For the latter, key presses are done in a row. Resulting in 18 different gestures – too few for the alphabet. Therefore, method 1 uses different *key maps* to approach the needed number of gestures. With two different key maps, users can switch between maps dependent on which map the needed character is located. Thus, each gesture is used twice, i.e. one time on each map similar to the "caps lock" on a QWERTY keyboard. Unlike upper and lower case versions of the same letter, changing the map results here in different characters. To minimize map switching activities, letters were grouped together based on their statistical occurrence in English, as some characters are more likely to occur together than others. For both methods, we considered the character distribution from [7]. Simple gestures that are quickly to execute and easy to memorize were deliberately mapped to often occurring characters. E.g., to enter a "t" a single gesture or a click on a button is needed.

Method 1			
Gesture	Button	Map1	Map2
rotLeft		DEL	DEL
rotRight		SPC	SPC
	B1	CHG	CHG
	B2	e	m
Up		t	g
Down		o	f
Up	B1	a	y
Down	B1	RET	.
rotLeft	B1	r	b
rotRight	B1	n	w
Up	B2	i	k
Down	B2	s	v
rotLeft	B2	h	x
rotRight	B2	l	z
Up	B1+B2	d	j
Down	B1+B2	c	q
rotLeft	B1+B2	u	
rotRight	B1+B2	p	

with B1 = Button1
and B2 = Button2

Method 2			
Gesture1	Gesture2	Button	Char.
		B1	e
		B2	t
Up			o
Down			a
rotLeft			DEL
rotRight			SPC
rotLeft	Up		h
rotLeft	Down		l
rotRight	Up		d
rotRight	Down		c
Up		B1	n
Down		B1	r
rotLeft		B1	u
rotRight		B1	p
rotLeft	Up	B1	f
rotLeft	Down	B1	y
rotRight	Up	B1	b
rotRight	Down	B1	w
Up		B2	i
Down		B2	s
rotLeft		B2	m
rotRight		B2	g
rotLeft	Up	B2	k
rotLeft	Down	B2	v
rotRight	Up	B2	x
rotRight	Down	B2	z
Up		B1+B2	j
Down		B1+B2	q
rotLeft		B1+B2	.
rotRight		B1+B2	RET

Figure 2: Gesture-to-Character mappings of method 1 and 2.

To delete a letter; an intuitive single "left-rotation" gesture is used. Figure 2 shows the complete gesture to character mapping table. Note, that with an additional key map numeric inputs could be allowed as well.

Method 2

The second method is using 30 gestures from the 34 gestures available as described in the beginning. Each letter in the alphabet has a distinct gesture. The remaining gestures are used for modifier keys. Figure 2 shows the used mapping table of method 2 where each letter is accessed by a distinct gesture or button click.

Experiment

By comparing method 1 with method 2 we want to investigate if there is a difference between the two methods and if method 1 allows faster typing and easier learning. We assume that by using easy to memorize gestures in combination with key maps, users are likely to perform better in terms of both typing speed and learning than with many direct but complex gestures. A user study was conducted to draw first conclusions.

Experiment Setup

20 subjects (avg. age 25.48) recruited from students of the local university and university unrelated persons participated in the study. The experiment uses a "between-subject" design with the input method as the single independent variable. Subjects were divided in two groups of equal size and were paid after completing the last session with 20 Euros. Each subject had an initial introductory session of 20 min. for explaining the study and input method, measuring their typing speed on a QWERTY keyboard, and to make them familiar with the glove and software. Afterwards, the first of 5 experimental sessions (20 min. each) using an input method was started. The remaining sessions were distributed over four directly following days. In total, each subject used its assigned method for 100 minutes over one week.

Figure 3: Subject performing a text input method.

To record data and to compute evaluation metrics, we altered the "Twidor" software [6] according to our needs. A visual help system was added that presents the gestures that belong to a requested input letter if a subject does not remember it within a time frame of 3 seconds (see figure 3). Subjects were requested to stand in front of a TFT-Monitor that shows the application to include the mobility aspect of wearable applications. An even more realistic setup using a head-mounted display for presentation was omitted to avoid nuisance factors caused by the novelty of the display.

Preliminary Results

Although the conducted experiment was rather small in terms of learning time, the preliminary results confirm our hypothesis that method 1 is superior to method 2.

The use of a distinct gesture for each character impairs learning and typing performance. A repeated measurements analysis shows a significant ($p < 0.05$) difference on the words-per-minute (WPM) rate.

As indicated in figure 4, subjects that used method 1 reached a higher WPM rate already after the first session. Considering the development of both curves they appear to diverge further if more session would be added; indicating that WPM rate is likely to increase with more learning time. The gradient of the WPM curve of method 1 is bigger than of method 2; indicating that method 2 is more difficult to learn or to use.

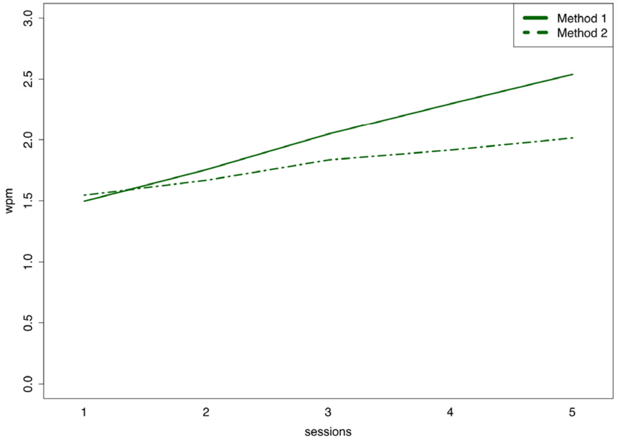

Figure 4: Average WPM rate achieved with both methods.

Figure 5 shows the average-error-rate (AER) of both methods that is deceasing for both over time. Unlike with the WPM rate where method 2 shows slightly better performance in the first session, the AER of method 2 is higher in all sessions. The AER curves, however, show that both methods can be learned. Subjects were able to avoid approx. 5% more errors with method 1 (29.36%) than with method 2 (34.30%).

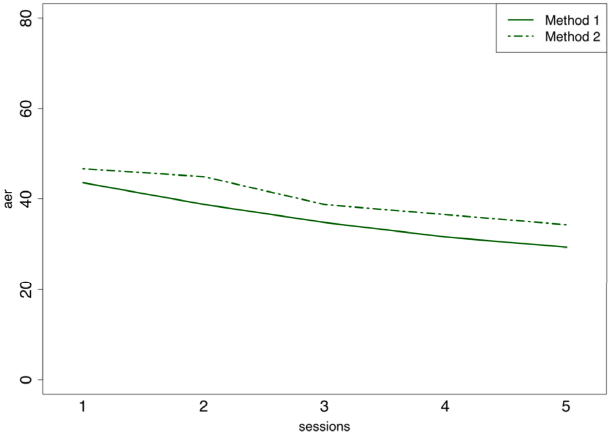

Figure 5: Average error rate of both methods in each session.

Although method 1 outperformed method 2 in WPM rate and AER, the absolute values achieved are rather low or high, respectively. With method 1, subjects reached 2.54 WPM on average with an AER of 29.36%. These values are deemed to be sufficient for short text input in professional wearable computing environments. The AER can be expected to decrease further in every day use.

Besides these statistical considerations, we observed some aspects during the experiment that allow an optimization of the input methods or should be considered in further calculations of performance metrics.

We found that some participants tried to correct every error, which slows down their input speed while only fewer participants did not correct any error; resulting in a bias for the small set of subjects and sessions that impaired the metrics used. Some of the participants always waited for the visualization of the gesture and apparently did not try to memorize them which cause the average WPM rate to be lower that it actually was. Here, the use of an improved "incentive scheme" applied for payment of subjects may help. Finally, we observed subjects having problems with the gesture recognition that sometimes did not correctly recognized entered gestures which impairs AER metrics. By using the recently proposed approach in [9] for calculating the AER this may be improved. However, also the gesture recognition has to be customized in particular with respect to chording. Currently, it requires some small idle time between gesture combinations that slow down typing speed and can be an additional learning burden or source for recognition errors.

Conclusion and Future Work

We presented two hand gesture based text input methods using a chording principle. Our study showed that a chording based input method together with a key map approach and a smaller number of simple and easy to memorize gestures outperforms a method where complex but distinct gestures for each letter are used. Within the given short learning time subjects were able to significantly increase their typing speed to a level that allows short text and abbreviation input needed in industrial wearable computing environments though.

Although first results are promising, data has to be examined further to uncover more optimization potentials. Moreover, another study that extends learning time and number of subjects involved is needed to determine long term properties of both methods with regards to learn ability and typing speed. Applying techniques such as T9 on top of our approach may yield insight on how far WPM rates can be raised.

References

[1] Bowman, D. A., Wingrave, C. A., Campbell, J. M., Ly, V. Q.: Using Pinch Gloves for both Natural and Abstract Interaction Techniques in Virtual Environments. *Proc.* HCI International 2001

[2] Byung Seok, and Yoon Sang Kim and Sang-Goog Lee: Improving the Usability of a Wearable Input Device – Scurry™, *Proc.* UbiComp 2004

[3] Doo, M., Lyons, K., and Starner, T. 2006. The Korean twiddler: one-handed chording text entry for Korean mobile phones. *Proc.* CHI 2006 Extended Abstracts

[4] Lee, S., Hong, S. H., and Jeon, J. W.: Designing a universal keyboard using chording gloves. *Proc.* ACM CUU 2003

[5] Liu, Y., Liu, X., and Jia, Y.: Hand-Gesture based text input for wearable computers. *Proc. IEEE* ICVS 2006

[6] Lyons, K., Starner, T., et al. Twiddler Typing: One-Handed Chording Text Entry for Mobile Phones. *Proc.* CHI 2004

[7] Rosenberg, R., Slater, M., The chording glove: a glove-based text input device, IEEE TSMC Part C, 2(2), p. 186-191, 1999

[8] Witt, H., Nicolai, T., Kenn, H.: Designing a Wearable User Interface for Hands-free Interaction in Maintenance Applications. *Proc.* PerCom 2006

[9] Wobbrock, J.O., Myers, B.A.: Analyzing the input stream for character-level errors in unconstrained text entry evaluations. ACM TOCHI 13(4), p. 458-489, 2006.