The Chording Glove: A Glove-Based Text Input Device

Robert Rosenberg and Mel Slater

***Abstract*—This paper introduces a new text input device called the chording glove. The keys of a chord keyboard are mounted on the fingers of a glove. A chord can be made by pressing the fingers against any surface. Shift buttons placed on the index finger enable the glove to enter the full ASCII character set. The chording glove is designed as a text input device for wearable computers and virtual environments. An experiment was conducted to assess the performance of the glove. After an average of 80 min of a tutorial, ten subjects reached a continuous text input speed of 8.9 ± 1.4 words/min, and after 10 1-hr sessions, they achieved 16.8 ± 2.5 words/min.**

***Index Terms*— Chord keyboard, disabled users, glove interfaces, input devices, one-handed keyboard, portable computers, QWERTY, text entry, virtual reality, wearable computers.**

I. Introduction

In this paper, a new glove-based text input device called the chording glove is introduced to provide a text input for a wearable computer. This uses chording to generate characters, similar to a chord keyboard, except that the keys are mounted on the fingertips of a glove and the chords are formed by pressing against any hard surface. There are clear space advantages over both a conventional keyboard and a chord keyboard. The purpose of this paper is to introduce the device and assess its performance. In Section II, the device is described in detail. Section III details a performance experiment, and the results are given in Section IV. Finally, in Section V, we speculate about the utility of the glove as a text input device for wearable computers.

II. The Chording Glove: A New Kind of Chord Keyboard

On a standard keyboard, a character is made by pressing either a single key or a key in conjunction with a shift or control key. The layout permits any number of characters, as long as there is room on the keyboard. A chord keyboard takes a different approach. There is one key for each finger. Multiple keys are pressed simultaneously in various combinations to enter characters, much in the same way that a chord is made on a piano. Pressing combinations of keys in this way is called *chording*.

The chording glove is a chord keyboard where the buttons have been mounted directly on the fingers themselves. The 31 character limit of a chord keyboard is surpassed by using three sticky shifts, mounted on the side of the index finger, in order to be pressed by the thumb. Function keys are provided to enable quick use of rarely used utilities.

Manuscript received May 18, 1996; revised August 23, 1998.

The authors are with the Department of Computer Science, University College of London, London WC1E 6BT, U.K. (e-mail: B.Rosenberg@cs.ucl.ac.uk; M.Slater@cs.ucl.ac.uk).

Publisher Item Identifier S 1094-6977(99)02768-6.

A. The Glove Layout

The chording glove (Fig. 1) consists of three parts.

1) finger sensors;
2) shift buttons;
3) function keys.

1) Finger Sensors: The finger sensors detect when a finger is pressed against something. Sensors are located at the tip of each finger, except for the thumb. The thumb's sensor is on the side, just below the knuckle. This facilitates chording when the hand is horizontal, as the side of the thumb is usually used.

2) Shift Buttons: There are three sticky shift buttons on the side of the index finger [Fig. 1(c)]. These are pressed either by the thumb or by the other hand. The shift buttons are used to change each shift state—*caps*, *number*, and *control*. A single press applies the shift to the next character made. A double-press locks the shift until it is pressed again. By using sticky shifts, the number of possible characters is increased from 31 to 248. LED's next to each shift show the state: a green LED for shifted and a red LED for locked.

3) Function Keys: The function keys are seldom used keys that are located on the back of the hand in easy reach of the other hand. These keys are pressed individually, and each has its own function.

`Pause` causes all finger sensors to be ignored until this key is pressed again. This is to allow the hand to perform other actions without accidental chording.

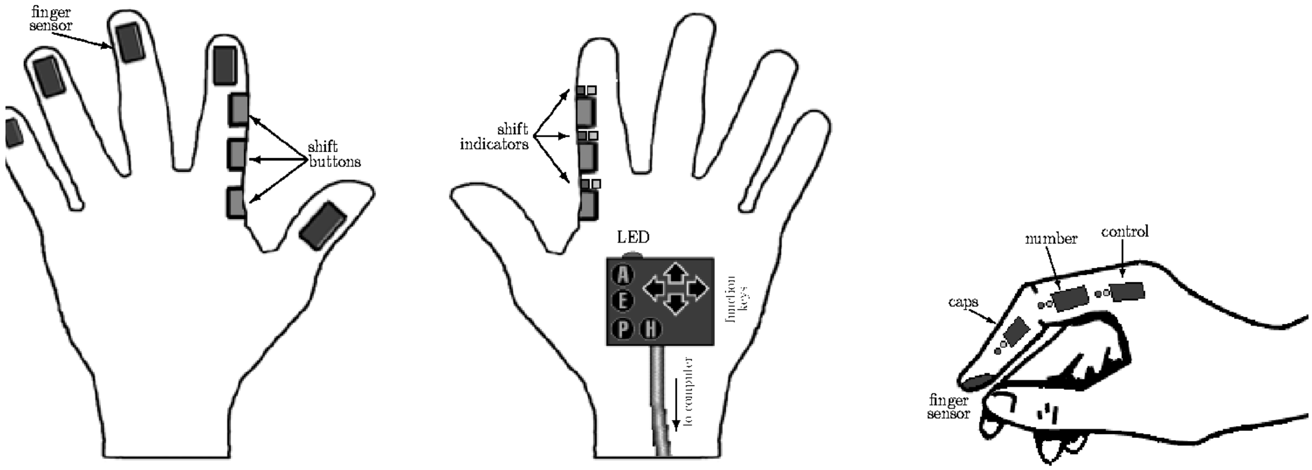
`Escape` has the same effect as on a normal keyboard.

A single press on `Help` calls an application sensitive help function, and a double press displays the chord keymap until the key is pressed again. This allows the user to be able to look up chords at any time.

`AutoCaps Toggle` turns on and off the AutoCaps feature. The AutoCaps feature automatically capitalizes the first letter of a sentence. This is to save effort of using the shift at the start of each sentence. The Caps shift turns on automatically whenever a sentence ending character is chorded: .!? (period, exclamation point, and question mark). There is an LED next to this button that shows if it is on or not.

There are four `Arrow Keys`.

1) $\uparrow$ (`Up`);
2) $\downarrow$ (`Down`);
3) $\leftarrow$ (`Left`);
4) $\rightarrow$ (`Right`).



(a) (b) (c)

Fig. 1. Chording glove. (a) Bottom view. (b) Top view. (c) Side view (the shift buttons)

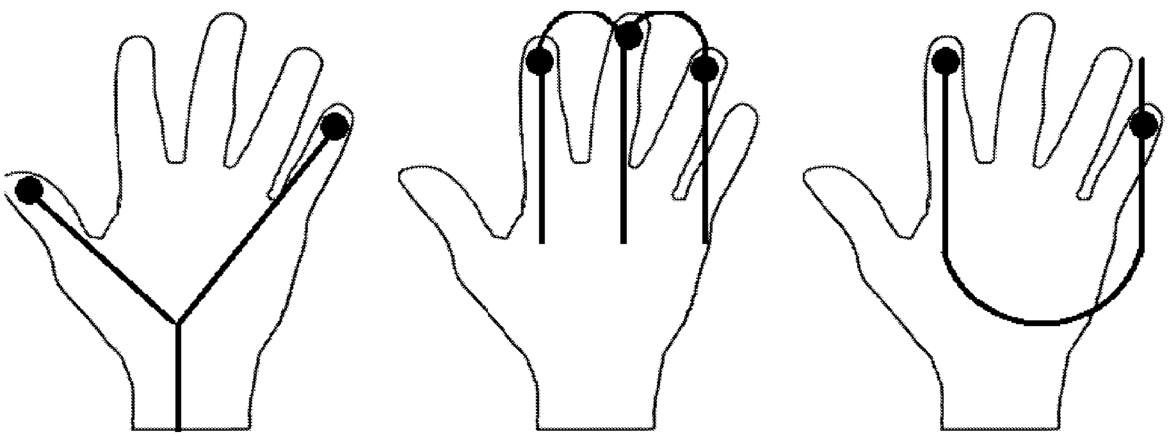


Fig. 2. Chords for the letters `Y`, `M`, and `U` are some of the chords that resemble the letters they make.

These act the same as arrow keys on a normal keyboard to provide basic pointer control.

B. The Chord Keymap

Seibel rated the relative difficulty of chords by measuring the discrimination reaction time (DRT) for each chord [1]. The chords with the fastest DRT's are the easiest to make. The chord keymap for the chording glove was developed by associating the most frequently used characters with the fastest DRT's. In addition, five basic mnemonics were used to facilitate memorization of the keymap.

1) *The chord can resemble the character.* The finger combination can have some obvious relation to the shape of the character typed. For example, the letters `Y`, `M`, and `U` are all made in such a way that the shape of the fingers which make the chord resemble the shape of the letters (Fig. 2).
2) *A sequence of chords can have some meaning and be easy to make*. For example, the common sequence `the` is made by a simple sequence of chords—Index → middle → ring—which are easy to make in succession.

TABLE I
CHORD PATTERN FOR NUMBERS

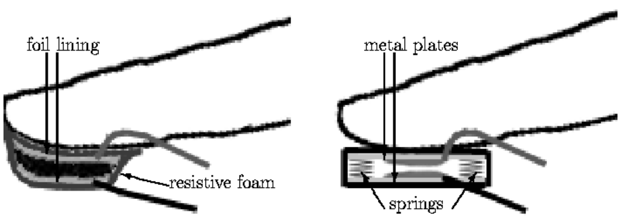
<table>
<tr><th></th><th>index</th><th>middle</th><th>ring</th><th>little</th></tr>
<tr><td>No thumb</td><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td>Thumb</td><td>5</td><td>6</td><td>7</td><td>8</td></tr>
<tr><td></td><td colspan="2">9</td><td></td><td></td></tr>
<tr><td></td><td colspan="3">0</td><td></td></tr>
<tr><td></td><td colspan="4">return</td></tr>
</table>

3) *Sequential characters can be made by following a simple pattern*. For example, numbers are made as described in Table I.
4) *One chord can be based on another chord.* For example, the similar characters `Comma` and `Period` have similar chords, with only one finger difference between the two (Table II).
5) *A shifted chord can be based on an unshifted chord.* For example, the chord for `exclamation point` is a shifted version of the chord for `e`.

This keymap was designed for the English language. However, it would be possible to redesign the keymap for other alphabets.

TABLE II
CHORD KEYMAP

Chord					Shifts			
Thumb	Index	Middle	Ring	Little	Default	Caps	Num	Caps+Num
●	●	○	○	○	a	A	5	&
●	●	○	●	○	b	B	*	
●	●	●	●	○	c	C	0	%
●	○	●	●	○	d	D	/	\
○	○	○	●	○	e	E	3	!
●	○	○	●	●	f	F	Tab	
○	○	●	●	○	g	G	)	>
○	○	●	○	○	h	H	2	@
○	○	○	○	●	i	I	4	?
●	●	●	○	●	j	J		
○	●	●	●	●	k	K	ˆ	ˆ
○	●	●	●	●	l	L	=	<
○	●	●	●	○	m	M	-	_
○	●	●	○	○	n	N	(	"
●	○	○	●	○	o	O	7	|
○	●	○	●	○	p	P	+	#
○	●	●	○	●	q	Q	?	
●	●	●	○	○	r	R	9	‘
●	○	●	○	○	s	S	6	$
○	●	○	○	○	t	T	1	'
○	●	○	○	●	u	U	[	{
○	●	○	●	●	v	V	Escape	
●	●	○	○	●	w	W	]	}
●	●	○	●	●	x	X	*	
●	○	○	○	●	y	Y	8	~
●	○	●	○	●	z	Z		
●	○	●	●	●	, (comma)			;
○	○	●	●	●	. (period)			:
●	○	○	○	○	Space			
○	○	○	●	●	Back Space			
●	●	●	●	●	Return			



(a) (b)

Fig. 3. Finger sensors. (a) Original “resistive foam” finger sensors. (b) Replacement “metal plate” finger sensors.

C. The Hardware

In the first version, the finger sensors were small, flexible, resistive foam sensors [Fig. 3(a)]. The sensor was triggered when the resistance across the foam dropped below a cutoff value. After the first few pilot experimental sessions, it was found that long term use caused increased noise levels, rendering the sensors useless after a few hours of chording. These were then replaced with larger, more reliable, metal plate buttons [Fig. 3(b)].

The new buttons were stable and much more accurate. The disadvantage was that they were much larger and more cumbersome to use. The large size most likely had a detrimental effect on accuracy and speed.

III. EXPERIMENT

The theoretical input speed for this keymap can be approximated by averaging Seibel’s DRT’s and weighting each DRT by the frequency of the associated character. This is a heuristic method for estimating the potential chording speed, and although the model is somewhat physiologically oversimplified, it does give a rough idea of speeds that might be expected. Using this method, a theoretical maximum speed for this keymap would be estimated as 305 ms/character or 40 words/min.

Seibel [1] calculated DRT’s by measuring the time it took for a subject to make a chord once asked to do so. The chords were displayed randomly and were not associated with any characters. When using chords for text entry, the following character is known in advance, giving the subject a slight time advantage, as entering text is always faster than entering random characters. This implies that it is possible to achieve faster speeds than the afore-mentioned theoretical 40 words/min. Further study must be carried out to show the validity of this model.

The experiment was designed to explore the following performance features of the chording glove:

1) Learning the Keymap: Chording glove’s 97 character keymap should be able to be memorized within 90 min: well enough to allow continuous text input.

2) Visual Supervision: Touch typing takes hours of practice to perform without looking. Even if the key layout is memorized, typists still tend to look at the keyboard when typing [2]. Users of the chording glove should be able to chord without needing to look at their hand and only need to look up infrequently used chords.

3) Portability: The chording glove should be able to chord on any surface without any significant loss in speed. This would allow the chording glove to be used anywhere without any loss in productivity.

4) Text Entry Rates: For beginning and moderate users, a chord keyboard should be as fast or faster than a QWERTY keyboard.

A. Materials

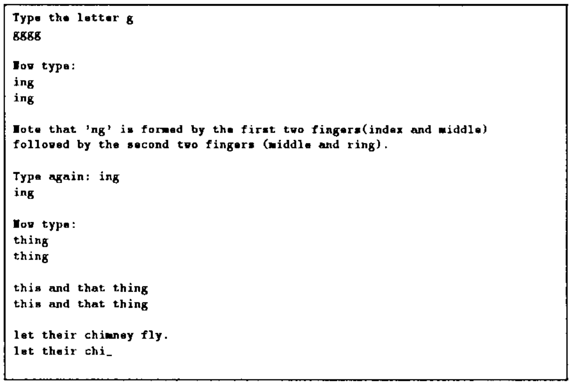
The chording glove was simplified for the experiment, limiting its features to those necessary for plain, line-by-line text entry with no editing functions.

1) Shift Buttons: The `control` button was not needed for the experiment. This button was remapped to be the `help` button. The keymap would be displayed for as long as the `help` button was held down. When the button was released the keymap would disappear (this was to measure how long the subject needed to view the keymap).

2) Function Keys: The experiment did not require any function keys, except for `help`, as described above. The function keys were removed from the glove to avoid any confusion.

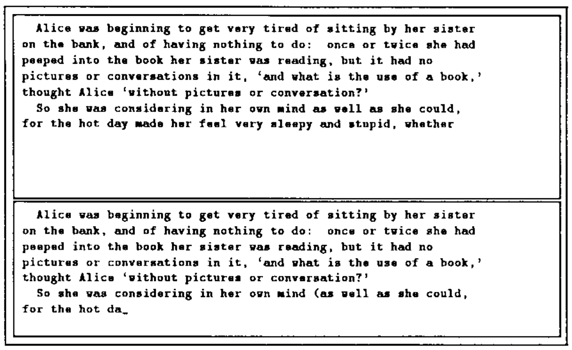
B. Method

1) Subject Selection: Ten subjects were selected from those who responded to advertisements around the college campus. There were four males and six females, and all were right handed and aged between 18 and 28 years. All the subjects described themselves as competent typists with six using



```
Type the letter g
gggg

Now type:
ing
ing

Note that 'ng' is formed by the first two fingers(index and middle)
followed by the second two fingers (middle and ring).

Type again: ing
ing

Now type:
thing
thing

this and that thing
this and that thing

let their chimney fly.
let their chi_
```

Fig. 4. Sample text from the tutorial session.



```
  Alice was beginning to get very tired of sitting by her sister
on the bank, and of having nothing to do:  once or twice she had
peeped into the book her sister was reading, but it had no
pictures or conversations in it, 'and what is the use of a book,'
thought Alice 'without pictures or conversation?'
  So she was considering in her own mind as well as she could,
for the hot day made her feel very sleepy and stupid, whether
```

```
  Alice was beginning to get very tired of sitting by her sister
on the bank, and of having nothing to do:  once or twice she had
peeped into the book her sister was reading, but it had no
pictures or conversations in it, 'and what is the use of a book,'
thought Alice 'without pictures or conversation?'
  So she was considering in her own mind (as well as she could,
for the hot da_
```

Fig. 5. Sample text from the chording session.

keyboards for only a few hours a week and the rest typing on a daily basis. No subjects reported ever having used a chord keyboard or having experienced any RSI. Each subject was paid £30.00 for completing the experiment. The subjects each performed ten chording sessions following a tutorial session spread out over a period of approximately two to three weeks.

2) Tutorial: The initial session was a tutorial of approximately one hour to teach the chord keymap (Fig. 4). For the tutorial the subject was given two sheets of paper, where the first listed the keymap (Table II), and the second displayed the chords, which are similar in shape to the characters they make. The subject learned each chord by being asked to chord a character and then generate several short phrases using it and some previously learned chords. The subjects could take as long as they needed to finish. Most took between 1 and 1.5 h to finish the tutorial.

3) Chording Sessions: There were ten chording sessions following the tutorial. The sessions consisted of 50 min of text input. Each subject was paid £1.50 at the end of each session. When the subject finished all the sessions, they were paid an additional £15.00.

For the chording sessions, the screen was divided into two windows (Fig. 5). The text that appeared in the top window was chorded by the subject and displayed in the bottom window. In each session, the subject was told to type as quickly and as accurately as possible. Correcting errors was not as important, and the subjects were told to fix mistakes only if it did not require too much effort.

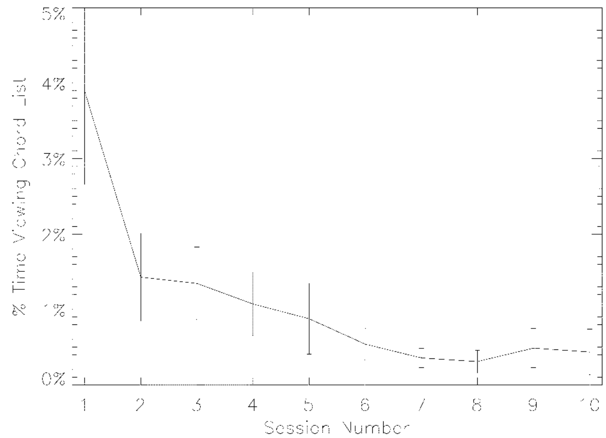


Fig. 6. Average percent time spent looking up chords per session.

Each session consisted of a 15-min trial, followed by a 10-min trial, another 15-min trial, and a final 10-min trial. The first and last were always text, whereas the second two alternated from session to session, being either text first, followed by data, or data first, followed by text. The variation in trial length and content was to avoid mental fatigue in the subject. After each chording session, the subjects were asked to fill out a questionnaire. The questionnaire asked to report their levels of fatigue, muscle strain, and what they liked, disliked, and what they would change about the chording glove.

IV. Results

A. The Chord Keymap

The subjects took an average of 80 min to complete the tutorial. By the end of this period, they were able to enter text while only rarely needing to view the keymap. This took, on the average, 1.25 h. This suggests that the chords had been learned well enough for continuous text entry. This is supported by the fact that less than 4% of the first chording session was spent looking up unremembered chords (this was the first session for which this data was available). By the last session, the amount of time spent looking at chords had dropped to 0.4% (Fig. 6).

B. Input Speed and Error Rate

At the end of the tutorial, the average overall speed was 8.9 $\pm$ 1.4 words/min. The error rate was calculated as the ratio of chording errors to the total number of characters. This was 27 $\pm$ 2.5% after the tutorial. The chording speed increased over the sessions with no signs of leveling off. The average chording speed of the final session was 16.8 $\pm$ 2.5 words/min. The final error rate had fallen to 17.4 $\pm$ 0.6% with some signs of leveling off (Figs. 7 and 8).

One subject was asked to return three and six months later for an additional chording session. After three months, at the start of the session, her input speed was 13.2 words/min, but this rose to 17.0 words/min within 20 min. Her chording speed had risen 5% since her tenth session. The subject spent 2.8%

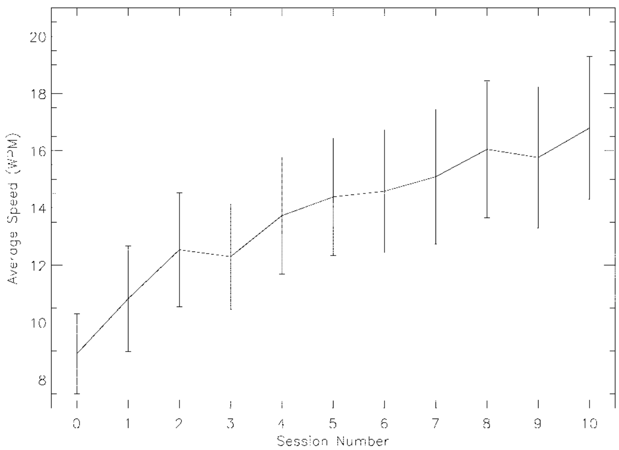


Fig. 7. Average chording speed for all subjects per session (in words per minute).

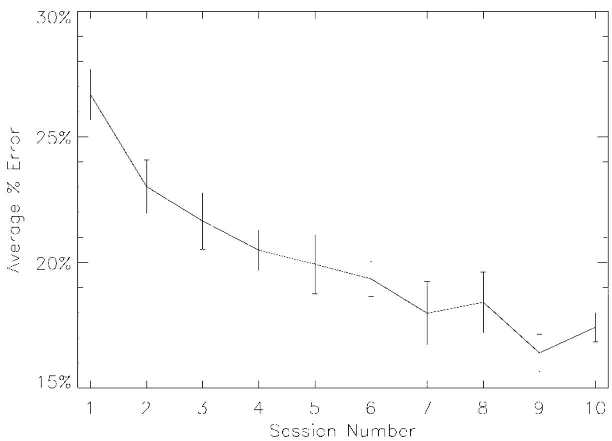


Fig. 8. Average percent error for all subjects per session.

of the time looking up chords. Three months later, at the start of the session, her input speed was 13.9 words/min and rose to 17.9 words/min within 30 min. This is an increase in chording speed of 10% from the tenth session. The subject spent 1.2% of the time looking up chords.

C. Fatigue

Fatigue was assessed in the questionnaire given after each session. Fatigue was measured by asking the subject if they felt that they could chord for longer, had chorded too much, or if they had chorded for the right length of time. The scale was 1 to 5, where 1 was too much chording, and 5 was too little. After the last session, the average value was 3.9 ± 1.0.

D. Portability

To measure portability, the subject was asked to type for a trial (text entry) while standing up. They were allowed to type on any surface they wished (a desk, a file cabinet, the computer monitor, etc.). The input speed of that trial was compared with the input speed of the other two text entry trials in that session. The average digram time (the time between one chord and the next) was calculated and compared with the average digram time while sitting. The difference in digram times was $+0.47 \pm 5.48$ words/min. There was, therefore, no significant difference in input speed while standing and while sitting.

V. Discussion

The subjects took an average of 80 min to learn the entire chord set well enough to allow continuous text entry. After 6 h of use, the subjects only needed to look up a few seldom-used chords. Most chord keymaps are claimed to take between 30 to 60 min to memorize [2]–[5]. The time for this keymap is slightly longer because the character set is much larger, containing all the characters from a standard keyboard. Most other chord keymaps used in these studies include only the letters and not numbers and punctuation.

After about 1 h, the subjects could chord without looking at a guide. Touch typists often require visual supervision long after the keyboard layout has been memorized. Many casual typists are never able to type without visual supervision [2]. This implies that it is faster to learn touch typing on this chording system than on a standard keyboard.

Most subjects reported feeling some pain in their hand when they first used the glove. This occurred for the first few sessions. By the last session, the subjects claimed to feel some pain immediately after typing, but this quickly diminished. None of the subjects reported pain in their upper arms or back during the course of the experiment: only the hand. However, as each session was only 1 h long, there is no data as to the effects of prolonged use of the chording glove. This needs to be addressed in further experimentation.

An additional positive benefit of the chording glove is that only one hand is required. Most subjects reported that they liked the fact that one hand was free to perform other tasks. The other most liked aspect was the fact that they never needed to look away from the screen when chording. There is no change in input speed for the chording glove even when the subject is in a position in which it would be very difficult to type on a keyboard. This has positive ramifications for the use of the chording glove for a mobile computer.

While only one subject was used to determine the long-term effects, the results implied that a user returning to the chording glove after a long absence can quickly recall the keymap. This implies that the keymap goes into long-term memory after the ten sessions. To give a fuller understanding of keymap retention, this experiment needs to be extended to include more users and testing over a longer period of time.

After 10 h of training, the average input speed for the chording glove was 16.8 words/min. The input speed on a QWERTY keyboard for a previously untrained user after 12 h of training is 20 words/min [6]. The slower input speed of the chording glove is due to the low quality finger sensors actually used in the experiment. They were not sensitive enough for general text input. In addition these sensors were also too large to allow comfortable freedom of movement. Smaller, more accurate sensors should increase the text input speed significantly.

The standard error rate for the QWERTY is 12.7% [7], which is slightly less than the chording glove's 17%. The inflated error rate can be explained by the low accuracy of the sensor. More accurate sensors should reduce the error rate to a more comparable level.

The fatigue the subjects felt lessened as they used the chording glove more. The final value was higher than desirable. Like error rate and input speed, this can be blamed on the finger sensors, which required more pressure than they should have to work. Smaller sensors would allow the users to chord in more comfortable positions, reducing strain. Lowering the noise and increasing the sensitivity of the sensors should reduce the amount of work in chording, reducing fatigue.

VI. Conclusion

The implementation of the chording glove used in the experiment was somewhat flawed in the bulky and inaccurate finger sensors. This led to a certain disadvantages as compared with the standard QWERTY keyboard, such as slower input speed. Any future models will have more accurate sensors, which should improve the chording glove's performance. Thus, the 16.8-words/min speed is a conservative estimate of the possible speed for the chording glove.

The evidence suggests that the keymap can be learned quickly, in a time comparable with other chord keyboards, and that the keymap can be retained in memory over long periods, but another experiment must be done to prove this. It has been shown that the chording glove enables the users to enter text without any visual supervision, and it may be possible to improve on this further by adding audio or tactile feedback. There is no significant difference in the performance of the chording glove when the user is standing or sitting; therefore, the chording glove might be used in a portable environment. However, further research needs to be done in more complicated situations, such as chording on a part of the body or while moving. A flexible capacitive membrane sensor would be the mostly likely candidate to improve the finger sensors. Other topics that should be looked into are the effects of adding function keys, the long-term effects on the muscles, and the effects of combining with a pointer input.

The chording glove has many potential uses. The compactness and flexibility for chording give it potential as an input for a wearable computer. Most wearable text inputs use a board braced between the hand and something else. Keyboard and stylus inputs must be held or strapped to the body to be used with a wearable computer. This usually takes up one or both hands, limiting the real-world tasks the user can do at the same time. Voice input devices avoid these problems and are very portable, but they have numerous privacy problems and may not work at all in a noisy area. The chording glove is completely passive and can be called into use at any time. When it is not wanted, it can be ignored, just as one can ignore wearing any normal glove.

The simplicity of the chording glove interface makes its use especially easy for visually impaired people. Learning to touch type on a chord keyboard is much easier than on a standard one. This could open up computing to many people who would have previously found it difficult or impossible.

The chording glove can be modified for use in a virtual environment. It would be simple to combine it with an existing virtual reality glove interface. This would expand the uses of immersive virtual reality to include all text aspects of modern desktop computers. This would allow one to program, write e-mail, and manipulate virtual objects without leaving the virtual environment.

Acknowledgment

The authors would like to thank H. Johnson and D. Beal for their comments on this project and earlier drafts of this paper. They would also like to thank D. Coppen and T. Barnes for technical assistance and all the subjects who participated in the experiment.

References

[1] R. Seibel, "Performance on a five finger chord keyboard," *J. Appl. Psychol.*, vol. 46, no. 3, pp. 165–169, 1962.
[2] D. Gopher and D. Raij, "Typing with a two-hand chord keyboard: Will the QWERTY become obsolete?," *IEEE Trans. Syst., Man, Cybern.*, vol. 18, pp. 601–609, July/Aug. 1988.
[3] A. Kirschenbaum, Z. Friedman, and A. Melnik, "Performance of disabled people on a chordic keyboard," *Human Factors*, vol. 28, no. 2, pp. 187–194, Mar. 1986.
[4] *BAT IBM-PC Quick Reference Guide*. Seattle, WA: Microsoft Corp., 1995.
[5] S. Roberts, "Chord keyboards," *Microship Status*, no. 80, Jan. 1995; Available WWW: http://www.microship.com/gopher-data/Microship_Status/Microship_Status_950131.
[6] J. Noyes, "The QWERTY keyboard: A review," *Int. J. Man-Mach. Studies*, vol. 18, no. 3, pp. 265–281, 1983.
[7] K. M. Potosnak, "Keys and keyboards," in *Handbook of Human-Computer Interaction*, M. Helander, Ed. New York: Elsevier, 1988, ch. 21, pp. 475–494.

Robert Rosenberg received the B.S. degree in engineering physics from Cornell University, Ithaca, NY, in 1993 and the Ph.D. degree from University College, London, U.K., in 1998, concentrating in alternative input devices for wearable computers.

Mel Slater is a Professor of Virtual Environments in the Department of Computer Science, University College, London, U.K. He was a Visiting Professor with the Computer Science Division, University of California, Berkeley, in 1991 and 1992 and a Visiting Scientist at the Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, during the Spring semester of 1998. He leads a number of projects in the area of shared virtual environments, and his main research interests are concerned with presence, copresence between participants in a shared virtual environment, and computer graphics rendering algorithms.