Expert Chording Text Entry on the Twiddler One–Handed Keyboard

Kent Lyons, Daniel Plaisted, Thad Starner
College of Computing and GVU Center
Georgia Institute of Technology
Atlanta, GA 30332-0280 USA
{kent, plaisted, thad}@cc.gatech.edu

Abstract

Previously we demonstrated that after 400 minutes of practice, ten novices averaged over 26 words per minute (wpm) for text entry on the Twiddler one–handed chording keyboard, outperforming the multi–tap mobile text entry standard. Here we present a study that examines expert chording performance. Our five participants achieved an average rate of 47 wpm after approximately 25 hours of practice in varying conditions. One subject achieved a rate of 67 wpm, equivalent to the typing rate of the last author who has been a Twiddler user for ten years. We analyze the effects of learning on various aspects of chording, provide evidence that lack of visual feedback does not hinder expert typing speed and examine the potential use of multi–character chords (MCCs) to increase text entry speed.

1 Introduction

Mobile computing is becoming one of the most widely adopted computing technologies. There are currently 1.3 billion mobile phone subscribers and could be as many as 2 billion by 2007 [1]. Wireless text messaging is widespread with predictions of a rate of over 1 trillion messages per year being reached shortly [4, 11]; unfortunately, slow text entry on mobile devices may limit the utility of upcoming services such as wireless e-mail. In this paper, we present an evaluation of a chording method of text entry on the Twiddler, a 3x4 button keypad that offers rapid typing rates. We explore the rates of learning for chording, present data on our expert participants using multi-character chords (MCCs), and examine the effects of varying visual feedback on expert typing speeds.

1.1 Twiddler Chording

Many wearable computer users [5, 14] type with the HandyKey Twiddler (Figure 1), a mobile one–handed chording keyboard with a keypad similar to a mobile phone. The Twiddler has twelve keys arranged in a grid with three

Figure 1. Chord for the letter 'j' (R0L0) on the Twiddler

columns and four rows on the front. The device is held with the keypad facing away from the user and each row of keys is operated by one of the user's four fingers. Additionally, the Twiddler has several modifier buttons such as 'Alt', 'Shift', 'Control', etc. on the top–back operated by the user's thumb. Instead of only pressing keys in sequence to produce a character as with traditional keyboards, multiple keys can be pressed simultaneously to generate a chord.

The default keymap for the Twiddler is shown in Table 1 and consists of single button and two button chords which are assigned in an alphabetical order. The four characters in the Buttons column denote what keys to press from each row. 'L' indicates the leftmost button in a row, 'M' the middle and 'R' the right button. A '0' means the corresponding finger is not used in the chord. The chord for 'a' is 'L000' which indicates that the user should press the left button on the top row. To generate 'j' ('R0L0'), the user would press the right key on the top row and the left key on the third row

(Figure 1). Note that the designation for left and right is from the user's perspective while holding the keypad facing away. As a result, there is a left–to–right mirror between Table 1 and Figure 1.

Buttons	Char	Buttons	Char	Buttons	Char
L000	a				
0L00	b	RL00	i	ML00	r
00L0	c	R0L0	j	M0L0	s
000L	d	R00L	k	M00L	t
M000	e				
0M00	f	RM00	l	MM00	u
00M0	g	R0M0	m	M0M0	v
000M	h	R00M	n	M00M	w
R000	Space				
0R00	Delete	RR00	o	MR00	x
00R0	Backspace	R0R0	p	M0R0	y
000R	Enter	R00R	q	M00R	z

Table 1. Keymap for chording on the Twiddler.

With traditional keyboards, a character is generated when the corresponding button is pressed. This strategy cannot be used for chording since the user may not press all of the keys for the chord at exactly the same time. Instead, the Twiddler generates the keycode once the first button of a chord is released. Just before this point, all of the buttons for the chord have been depressed so the proper keycode can be generated. In Section 2.2, we explore the relationship between the timings of pressing the buttons and how they relate to learning to chord.

For a chord on the Twiddler, each of the fingers may be in one of four states (pressing one of three buttons, or not pressing anything). Ignoring the "chord" in which no buttons are pressed, there are then $4^4 - 1 = 255$ possible chords using the four main fingers. The modifier buttons operated by the thumb allow more chords. HandyKey includes what we have termed multi–character chords (MCCs) in the default keymap: single chords that generate a sequence of several characters. For instance, there are chords for some frequent words and letter combinations such as "and", "the", and "ing". Users can also define their own MCCs. We present an evaluation and analysis of the effects of MCCs on expert typing rates in Section 3.1.

1.2 Previous Work

In our previous work, we evaluated the relative learning rates of typing with multi–tap versus typing with chording on the Twiddler [6]. We conducted a longitudinal study with ten participants. None of the participants had any experience with typing chords on the Twiddler. However, they had varying levels of practice typing with multi–tap.

The experiment was a 2 x 20 within–subjects factorial design in which we presented the participants with two conditions (multi–tap and chording) during 20 sessions of typing. A session consists of two parts delineated by typing condition and a five minute break in the middle. Each part of the session, which lasts 20 minutes, consists of several blocks of trials. A block contains ten text phrases of approximately 28 characters each which were selected randomly from the set of 500 phrases developed by MacKenzie and Soukoreff [9]. These are phrases specifically designed as representative samples of the English language. The phrases contain only letters and spaces, and we altered the phrases to use only lower case letters and American English spellings. The software used for our experiments (Figure 9) is designed to prompt the participant with the phrase to be typed and record the response and timings for all of the buttons pressed.

We found the mean entry rates for our ten participants for session one were 8.2 wpm for multi–tap and 4.3 wpm for chording. As sessions continued, the means improved and reached 19.8 wpm for multi–tap and 26.2 wpm for chording by the end of the study (20 sessions, 400 minutes). While both conditions showed improvement, the typing rates for the chording condition rapidly surpassed those of multi–tap (Figure 2). After 20 session it is clear that the learning for multi–tap has tapered off. As the regression curves indicate, there is minimal improvement with each additional session. Chording, however, is still showing strong signs of learning.

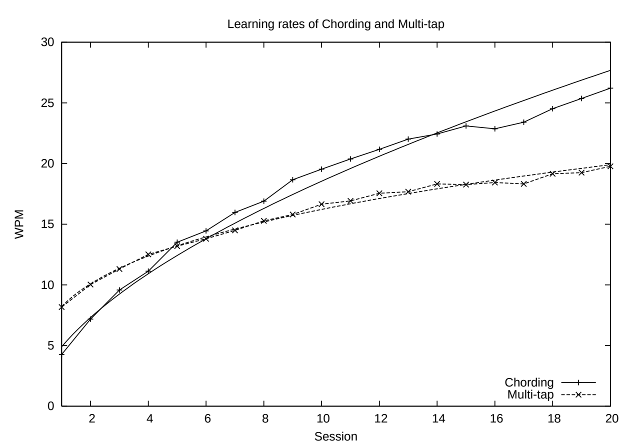


Figure 2. Learning rates and exponential regression curves for multi–tap and chording for 20 sessions [6].

Compared to other studies on mobile text entry using similar keypads using a 3x4 grid of buttons, our results are very encouraging (Table 2). Other studies have found that many text message users type near 8 wpm using multi–tap and T9 [2, 8]. Some experts can reach higher rates, but even new research text entry methods peak at 21 wpm [8]. In contrast after 400 minutes our participants reached an average typing rate of 26.2 wpm and were still improving.

In this paper, we present a follow–up study designed

Method	Keyboard	Experience	WPM
Chording [5]	Twiddler	400 min	26.2
LetterWise [8]	desktop keypad	550 min	21.0
T9 [2]	Nokia 3210 phone	expert	20.36
Multi–tap [8]	desktop keypad	550 min	15.5
T9 [2]	Nokia 3210 phone	novice	9.09
Multi–tap [2]	Nokia 3210 phone	novice	7.98
Multi–tap [2]	Nokia 3210 phone	expert	7.93

Table 2. Comparison of mobile text entry rates using 3x4 keypads.

to determine what chording rates our participants could achieve and to confirm or refute the expert rates predicted by our regression curves. We also analyze the nature of how the participants learned to type with chords. Finally, we examine the use of multi–character chords by our now expert typists and the effects of limited visual feedback.

2 Learning to Chord

The study presented here continues with a very similar procedure as in our previous work. For this study, five of our original ten participants agreed to continue and we resumed testing after a two week intermission. The five other participants from our original study declined to participate due to the large additional time commitment required. Our procedure was modified to focus our study on chording; we replaced the multi–tap condition from our original experiment with a second chording phase. As a result, each time a participant came in we collected two 20 minute sessions of chording data. For this experiment, we compensated each participant at the rate of \$0.33 $\times$ words per minute $\times$ accuracy.

2.1 Towards Expertise

The first portion of our study is designed to confirm the regression curves from our previous work which indicate the predicted learning rates for our participants. We collected data for approximately 20 additional sessions resulting in a total of 40 sessions or about 13 hours of practice per participant. We ended this phase when our participants were showing signs of expertise indicated by reduced rates of learning. Figure 3 shows the average typing speed across participants. Also plotted is the original regression from our first study and a modified regression based on the new data from our five participants. The dip in the typing rate at session 20 is the effect of the two week break between our original study and this follow up. While there was a decrease, the participants rebounded by the next session.

$$\text{Original regression :} \quad y = 4.8987x^{0.5781}, R^2 = 0.9849$$
$$\text{Modified regression :} \quad y = 5.3503x^{0.5280}, R^2 = 0.9787$$

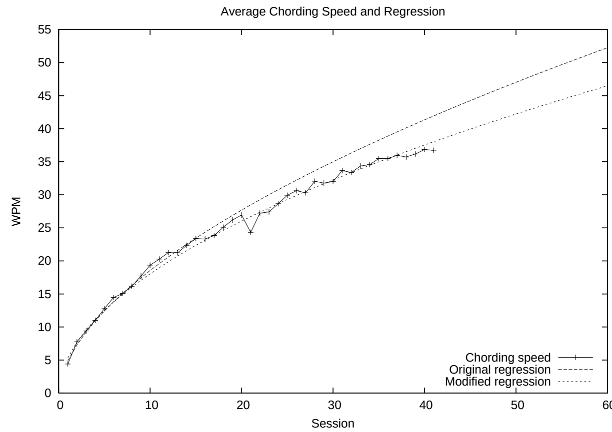


Figure 3. Mean learning rates and regression curves across participants.

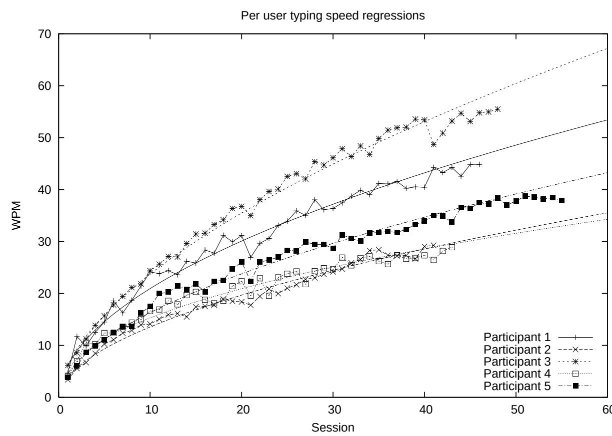


Figure 4. Per user typing rates and regressions

After 40 sessions the average typing rate for our participants increased to 37.3 wpm. This data shows that our original regression curve was slightly optimistic, predicting instead an average typing rate of 41.3 wpm. The difference could be a result of the variance in individual typing rates. Even though our regression fit to the mean typing rate of the participants is good, there are large differences in each individual's typing rate. Figure 4 shows the typing speeds for each of the participants by session. Also plotted are individual regression curves which have correlations of at least 0.96, indicating the data is well–fit.

Figure 5 shows the average error rate across participants using Soukoreff's and Mackenzie's total error rate metric [13]. The metric accounts for both corrected and uncorrected errors made by the participants providing a single total error rate. The final mean error is 6.2% and is slightly above other typing studies with a similar experimental design [8]. As shown, participants rapidly reduce their error rates as they initially learned to chord. As they learn to type faster, their accuracy gradually decreases. We believe this is

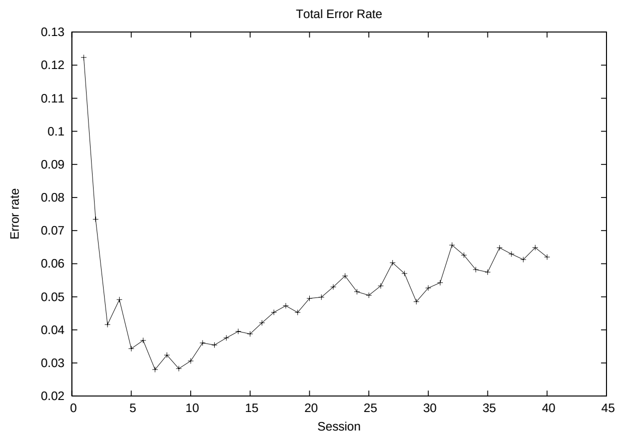


Figure 5. Mean error rate across participants.

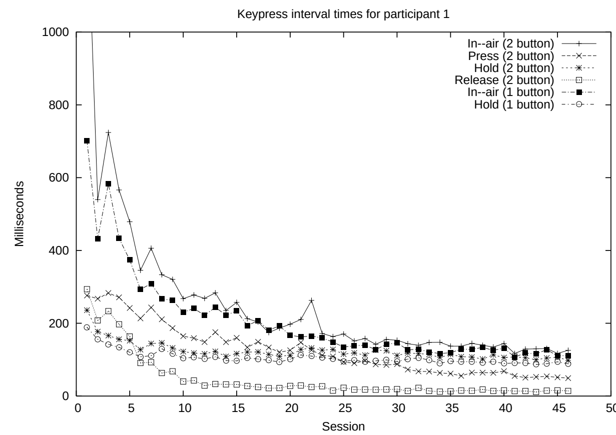


Figure 6. Keypress interval times for a single participant

an artifact of our experimental design as we did not directly control for accuracy. Instead, each participant was compensated proportional to the product of his rate and accuracy. As a result, the participants were rewarded if a small decrease in accuracy enabled a faster typing rate. A similar effect, where error rates gradually increase as participants become experts, was shown by Matias *et. al* with the Half–QWERTY keyboard [10].

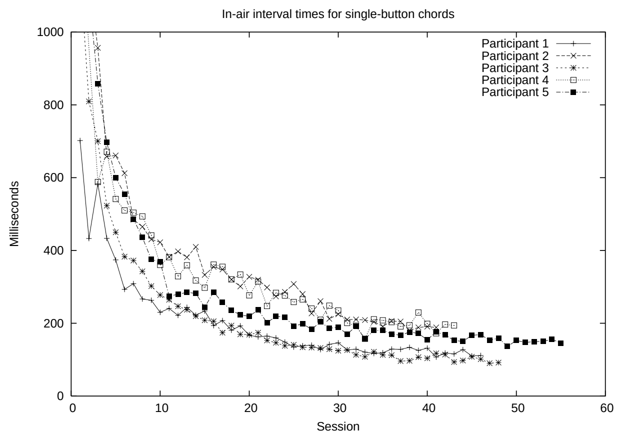
2.2 Analysis of Learning Rates

In addition to confirming the learning rate for the Twiddler, our additional data allows us to examine how users type on the Twiddler and to study the nature of the learning involved with chording. With a traditional keyboard, a character is generated by pressing and releasing a single key. Chord typing, however, may involve pressing and releasing two or more buttons to generate a character. We instrumented our experimental software to record the time each button is pressed and released for every chord. By examining the time intervals between each button press and release, we can gain insight into how novice users spend most of their time while learning and what optimizations we might make to aid expert users.

Typing a degenerate chord involving only a single button has one press and one release. This keypress has two intervals associated with it, in–air and hold. The first interval, in–air, is the time from when the last chord was completed (all of the buttons were released) to when the button for the current chord is depressed; in other words, the time when no keys are being held down. The other interval is the hold time and represents the interval between the press of the button and its release. We extended this notion of intervals to two button chords as well. The interval during which no buttons are pressed down is the in–air time, and the time during which all of the buttons are depressed is the hold time. However, the buttons in the chord may not be pressed or released at the exactly the same moment in time. This introduces two additional intervals. The time between the press of the first and second buttons of a chord is the press interval while the time between releasing the first and second button of a chord is the release interval. Thus, the sequence of two button chord time intervals is in–air, press, hold, and release, whereas single buttons only have in–air and hold intervals.

Figure 6 shows per–session averages of these intervals for a representative participant. This graph highlights where users spend their time in chording and suggests where the improvements of learning have the most effect. These values were computed by taking the intervals for each chord typed in sentences without any errors and then averaged for the whole session on a per user basis. We did not include sentences with errors as we did not want to confound our data on intervals. Mistyping one chord can impact several others, and it is not straightforward to incorporate the error data with our individual time intervals. We intend to examine errors and their effects more thoroughly in future work.

All of the participants' average in–air intervals for single and two button chords is shown in Figure 7. These time intervals exhibit the largest effects of learning. For novices, it is likely that this interval is dominated by the cognitive effort associated with remembering how to type each character and how to move their fingers to the correct position to type the letter. For experts, the delay becomes dominated by the time it takes to move the fingers from one chord to another. Comparing the in–air interval for single and two button chords reveals that, on a per user basis, the single button times are slightly faster and show better rates of learning. However, the two button in–air interval tracks the single button interval rather well. By the end of the study, the difference between the times on a per user basis becomes much smaller. On average our participants take 244ms to type a single button chord and 354ms for a two button chord. The discrepancy is mostly due to a single participant (number two) who is lagging behind on learning the



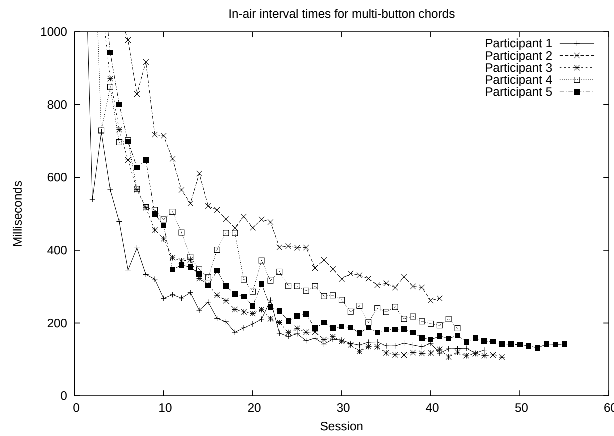


Figure 7. In–air interval times for single button chords (left) and two button chords (right).

two button chords. With additional practice his rates would approach the others and the difference between the in-air times for single and two button chords would decrease.

Figure 8 presents the press interval, which is the time between the first and second buttons of a chord being pressed. This interval is particularly interesting because it reveals different typing strategies between users. A single participant (number 3) always pushes both of the buttons in a chord at nearly the exact same time. The average delay between the first and second button press is only 7.25ms indicating that he always presses both buttons as one action. The other participants show a larger delay between these button presses, indicating that they press the buttons sequentially and likely learned how to press the chords in a different way than participant 3. The delay could be from planning and executing the two button presses in the chord separately. The slower users may also initially wait for haptic feedback from pressing the first button. For these participants there is some learning associated with this interval; however, it is not nearly as pronounced as the in–air time interval learning.

Participant 3 was significantly faster than the other participants and was typing at 67 wpm by the conclusion of our experiments. To see if this might be attributable to his simultaneous press strategy, we examined the data from the other five participants from our original study, who had stopped after 20 sessions. Two of the subjects employed the simultaneous press strategy, two of them the sequential strategy, and one started out sequential but appeared to switch mostly over to the simultaneous strategy by the end of the twenty sessions. The participants who used the simultaneous press strategy were no faster than those who used the sequential strategy. While simultaneous pressing might not produce the fastest rates while learning, it should be very beneficial for experts. At 60 words per minute, the average time to type one character is 200ms. Since the press

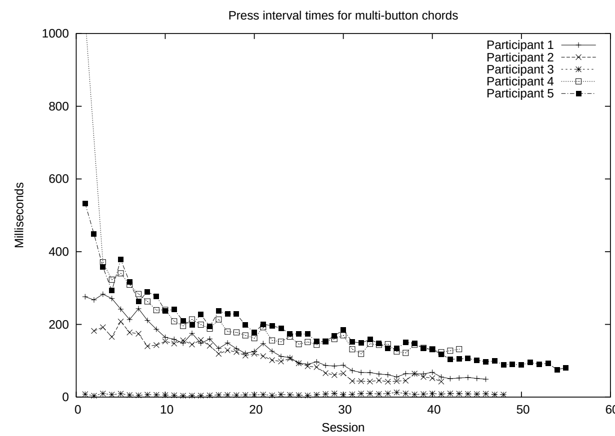


Figure 8. Press interval times (two–button chords)

interval times varied up to 100ms by the end of this phase and apply to more than 66% of the alphabet, pressing both buttons of a chord at the same time should significantly increase the typing rate.

Our last two times are the hold interval and the release interval (not shown). The average hold interval shows slight improvement with practice, and in general single button chords are held for slightly less time. At the end of this phase the single button chords are held 98ms while two button chords are held 107ms. Perhaps participants spend the extra time to ensure that they avoid releasing the first finger before the second one is depressed. Finally, while only one participant pressed both keys of a chord simultaneously all of the participants rapidly learned to release both buttons of a chord at approximately the same time. After about 10 sessions most of the users release both keys in less than 25ms.

3 Expert Usage

After about 40 sessions, enough data had been collected that we could be confident of our regressions' predictions.

While performance was still improving, the rate of learning had decreased enough that we considered our participants to be expert users. At this point we continued our experiment with two additional phases designed to investigate various aspects of expert typing. We examine the possible benefits of multi–character chords (MCCs) and the effects of typing with reduced visual feedback (blind typing).

3.1 Multi–Character Chords

As mentioned previously, there are 255 possible chords that can be typed on the Twiddler using the four fingers. Of these, only a small subset are allocated to the alphabet and punctuation needed to type English text. Some of the unused chords can be employed as multi–character chords (MCCs) which could generate any sequence of characters. In the next phase of our experiment we wanted to determine if MCCs for short common words and suffixes would improve our participants' typing rates. Our hypothesis is that MCCs would have a positive impact on typing rate because the number of button presses needed to type any given MCC string, such as "the ", would be reduced down to one chord. Using a MCC would reduce the overall number of keystrokes per character (KSPC) [7] as fewer keystrokes (button presses) would be needed to generate the same text.

We chose to investigate the benefits of MCCs by selecting 12 strings of at least three letters that are very prevalent in written English. The string were chosen using word frequency data from the commonly used text corpus, the British National Corpus [3]. For this experiment we selected 'for', 'and', 'the', 'ent', 'ing', 'tion', 'ter', 'was', 'that', 'his', 'all', and 'you' to be typed as MCCs. We assigned these strings to unused chords that did not involve the index finger. As many of these strings are normally followed by a space character, this assignment enabled us to add 12 extra MCCs that had a trailing space such as "the ". The buttons used for these chords are the same as the normal version, only the user also depresses the button used for space (the right button operated by the index finger). Table 3 shows the keymap for the additional MCCs.

Buttons	String	Buttons	String
0LL0	'for'	RLL0	'for '
0MM0	'and'	RMM0	'and '
0RR0	'the'	RRR0	'the '
00LL	'ent'	R0LL	'ent '
00MM	'ing'	R0MM	'ing '
00RR	'tion'	R0RR	'tion '
0LLL	'ter'	RLLL	'ter '
0MMM	'was'	RMMM	'was '
0RRR	'that'	RRRR	'that '
0L0L	'his'	RL0L	'his '
0M0M	'all'	RM0M	'all '
0R0R	'you'	RR0R	'you '

Table 3. Keymap for new multi-character chords (MCCs) with and without trailing space.

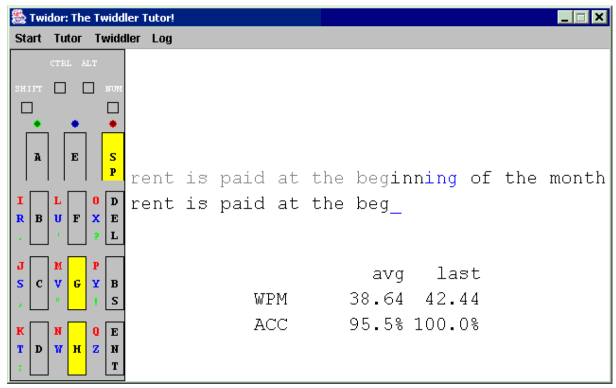


Figure 9. Our experimental software showing the use of MCCs; "ing " is the MCC to be typed ('R0MM') and is highlighted in blue.

To introduce MCCs to our participants, we modified the experimental software to highlight the next MCC that could be typed. Our software has a diagram of the Twiddler keypad that acts as a guide to help the users learn the basic alphabet keymap. We modified the diagram so that the keys needed for the MCC are also highlighted (Figure 9). We instructed our participants to use the MCCs and to encourage their use, we modified the error calculation so that typing the MCC string letter–by–letter counted against the participant's accuracy.

The effect of MCCs on our participants' typing rates is mixed. Initially, our participants slowed down as they started to learn MCCs. For the first session, the average typing speed dropped to 83.5% of what it had been. However on the fifth session, the average speed was 97.1% of the pre–MCC speed, and by the tenth session it was 104.5% and continued to improve. Even though the rate increased beyond the typing speed just before the introduction of MCCs, the participants were still slowly learning. If we had not introduced MCCs and just had our participants continue to practice, we would have expected the rate to increase to approximately 112% based upon our regressions. As a result we cannot attribute the overall increase in typing rate solely to the effects of MCCs.

To better understand the effects of MCCs we compared the amount of time participants needed to type a new multi–character cord compared to the time to type the chords for the alphabet which they already knew. In general, our multi–character chords might be slower because they involve more buttons (up to four) while the chords for the alphabet require at most two buttons. At the end of the MCC phase of our experiment, our participants were taking an average of 596ms to type each MCC, while two button alphabet chords took only 354ms and single took 244ms. We anticipate that once our participants mastered typing the new multi–character chords, the time per chord would drop and increase the overall typing rate. Looking at the time to generate the same string with and without the use of MCCs is also interesting. As mentioned, typing a MCC takes an average 596ms; while in contrast, typing out the same strings letter–by–letter takes 1018ms. As a result, there is a net savings in time using MCCs because typing the sequence with regular chords takes longer than typing the one multi-character chord.

An analysis of our phrase set revealed that 17.5% of the characters in our phrase set can be typed with MCCs. Weighted by the frequency of MCCs in our phrase set, this would correspond to about an 8% increase in average overall typing speed. This effect would likely be more pronounced using a phrase set more representative of English on a word frequency basis instead of letter frequency [9] and as our participants master the new multi–character chords.

3.2 Blind Typing

In a mobile environment, a user's visual attention may be diverted away from her display while entering text. Silfverberg examined the effect of visual and tactile feedback when using a mobile phone keypad [12]. He found that limited visual feedback combined with low tactile feedback hinders a user's average error rate; on the other hand, good tactile feedback results in much a smaller decrease in accuracy.

Inspired by these results and our own anecdotal experience of typing with limited visual feedback, we designed the last phase of our chording experiment to evaluate blind typing on the Twiddler. We have a 3 x 5 design with 3 conditions (normal feedback, dots feedback, and blind) over 5 sessions of typing. Each condition lasts 15 minutes. Our normal feedback condition displays the text typed under the phrase presented to the participant as shown in Figure 9, but without MCC highlighting. As the Twiddler is held with the keypad facing away from the user, this condition corresponds most closely to Silfverberg's indirect visual feedback condition. For our dots condition, we display periods for each character typed instead of the transcribed text. Thus, participants see their position in the supplied phrase, but not specifically what they type. This condition is designed to simulate monitoring text typed without being able to actually read the letters such as seeing the text on a heads–up display using only peripheral vision. Finally, the blind condition does not show any on–screen indication of what is typed and mimics Silfverberg's no visual feedback condition. For both the dots and blind conditions, participants are shown their transcribed text and error statistics when they press enter at the end of the phrase. We predicted that like Silfverberg, reducing the visual feedback would limit our participants' typing rate and accuracy.

Typing Rates (wpm)					
Participant	1	2	3	4	5
Normal	51.8	37.6	64.2	36.2	41.8
Dots	51.7	37.5	**67.2**	36.0	**43.1**
Blind	**53.7**	37.5	**67.7**	36.6	41.7

Percent Errors					
Participant	1	2	3	4	5
Normal	5.61	5.62	7.01	9.83	6.64
Dots	**4.82**	5.02	**5.75**	9.26	5.83
Blind	5.03	**4.63**	**5.90**	8.89	**5.44**

Table 4. Per participant typing and error rates for the three conditions. Bold indicates a statistically significant difference at the 0.05 level between that condition and the normal condition for that user.

Surprisingly, changing the visual feedback did not hinder the participants in their typing as expected. In some cases typing rates and error improved with the reduced feedback. Table 4 shows the change in speed and the error rate for the blind typing conditions. Values where a two–tailed t–test showed a statistically significant difference at the 0.05 level from the normal condition are marked with bold. Whenever there is a statistically significant difference between normal typing and one of the reduced feedback conditions, the reduced feedback condition shows an improved typing rate or a reduced error rate. More work will be needed to explore which factors affected this result.

3.3 Expert Typing Rates

By the end of all of our experiments, our participants completed an average of 75 sessions which corresponds to approximately 25 total hours of practice. Figure 10 shows the typing rates for our participants across all of our experimental conditions described above. The final average typing rate reached 47 wpm and unexpectedly our fastest participant achieved a rate of 67.1 wpm which is fast as the third author, an expert of ten years.

4 Future Work

In the future, we would like to create a model of Twiddler chording which accounts for finger motion and effects between chords. Our analysis of learning rates from Section 2.2 is a first step. Ideally, this model would enable us to evaluate different keymaps and optimize them for various tasks such as maximizing expert performance or easing learning. We also want to continue our study of multi–character chords to determine their effect on overall learning and typing speed and further examine the types of errors made while chording. Finally, we are interested in increasing the appeal of the Twiddler for novice users. We are developing a tutor to instruct people how to chord and eval-

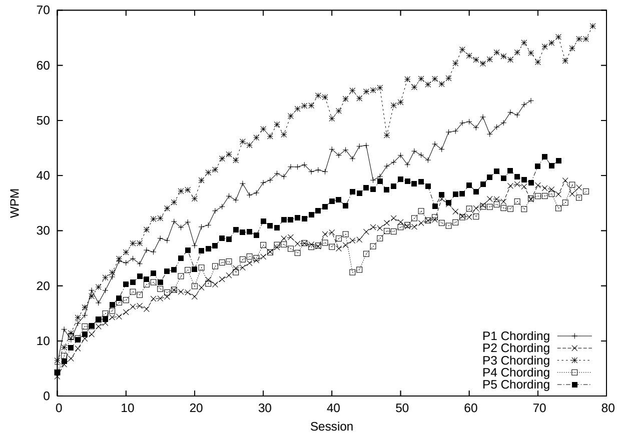


Figure 10. Data across all phases of experiment for all 5 participants.

uating the best mechanisms to teach chording. While the Twiddler offers very fast expert typing rates, we believe a tutor may improve the initial typing rate so that it is as fast as other mobile typing methods such as multi–tap. We are also exploring incorporating Twiddler style chording into a mobile phone that would offer better mass appeal.

5 Conclusion

We have analyzed various aspects of expert chording on the Twiddler keyboard including text entry speed, the effects of visual feedback, and the use of multi–character chords. We found that our participants reached an average typing rate of 47 wpm while our fastest participant reached 67 wpm. Our data on multi–character chords indicates that they could provide even higher typing rates. We examined how our participants learned to chord, showing most of the speed increase associated with learning occurs during the in–air time interval. We also found a difference in strategy of how our participants press the buttons of a chord. The blind typing data shows that the Twiddler can be used effectively with limited visual feedback which is important in a mobile environment. Given the expert users' high text entry speeds and ability to touch type, chording seems to be a viable mechanism for text entry on future mobile devices.

6 Acknowledgements

This work is funded in part by NSF Career Grant #0093291 and the NIDRR Wireless RERC. We would like to thank S. MacKenzie and W. Soukoreff for offering their phrase set for use, and S. Zhai and M. Silfverberg for their comments.

References

[1] S. Baker, H. GreenWith, B. Einhorn, M. Ihlwan, A. Reinhardt, J. Greene, and C. Edwards. Big bang! BusinessWeek, June 2004.

[2] C. L. James and K. M. Reischel. Text input for mobile devices: comparing model prediction to actual performance. In *Proceedings CHI*, pages 365–371. ACM Press, 2001.

[3] G. Leech, P. Rayson, and A. Wilson. *Word Frequencies in Written and Spoken English: based on the British National Corpus*. Longman, 2001.

[4] M. Lindstom. Message madness our big chance. SMH http://www.smh.com.au, February 2002.

[5] K. Lyons. Everyday wearable computer use: A case study of an expert user. In *Proceedings of Mobile HCI*, pages 61–75, 2003.

[6] K. Lyons, T. Starner, D. Plaisted, J. Fusia, A. Lyons, A. Drew, and E. Looney. Twiddler typing: One-handed chording text entry for mobile phones. In *Proceedings CHI*. ACM Press, 2004.

[7] I. S. MacKenzie. KSPC (keystrokes per character) as a characteristic of text entry techniques. In *Proceedings of Mobile HCI 2002*, pages 195–210, 2002.

[8] I. S. MacKenzie, H. Kober, D. Smith, T. Jones, and E. Skepner. Letterwise: prefix-based disambiguation for mobile text input. In *Proceedings of UIST*, pages 111–120. ACM Press, 2001.

[9] I. S. MacKenzie and R. W. Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI '03 extended abstracts*, pages 754–755. ACM Press, 2003.

[10] E. Matias, I. S. MacKenzie, and W. Buxton. One-handed touch typing on a qwerty keyboard. *Human-Computer Interaction*, 1996.

[11] Mobile CommerceNet http://www.mobile.seitti.com, January 2002.

[12] M. Silfverberg. Using mobile keypads with limited visual feedback: Implications to handheld and wearable devices. In *Proceedings of Mobile HCI 2003*, pages 76–90, 2003.

[13] R. W. Soukoreff and I. S. MacKenzie. Metrics for text entry research: an evaluation of msd and kspc, and a new unified error metric. In *Proceedings of CHI*, pages 113–120. ACM Press, 2003.

[14] T. Starner. *Wearable Computing and Context Awareness*. PhD thesis, MIT Media Laboratory, Cambridge, MA, May 1999.