

A Model of Two-Thumb Chording on a Phone Keypad

Nirmal Patel, James Clawson and Thad Starner
School of Interactive Computing and GVU Center
Georgia Institute of Technology
Atlanta, GA 30332
{merik,jamer,thad}@cc.gatech.edu

ABSTRACT

When designing a text entry system for mobile phone keypads, a designer needs to overcome the ambiguity that arises from mapping the 26 letters of the roman alphabet to only 12 keys (0–9, *, #). In this paper, we present a novel two-thumb chording system for text entry on a standard 12-key mobile phone keypad and introduce a performance model based on Fitts' Law for an expert user. The model provides a behavioral description of the user and predicts a text entry rate of 55.02 wpm.

Keywords

Mobile Text Entry, chording, Fitts' Law

Categories and Subject Descriptors

H.5.2 [Information Interfaces and Presentation]: User Interfaces
- Input devices and strategies, Interaction styles

1. INTRODUCTION

The use of text messaging services has become a very popular method for communicating when using mobile phones. Text messaging allows a user to asynchronously send a 160 character message to anyone with a mobile phone. It is estimated that over 1 trillion text (SMS) messages were sent in 2005.¹ The majority of today's mobile phones are equipped with at least one of the following input methods: a mini-QWERTY keyboard, a touch screen, or a 12-key mobile phone keypad. While phones with mini-QWERTY keyboards and touch screens are becoming popular, the 12-key mobile phone keypad is still the most used input device for mobile phones as shown in Figure 1

A mobile phone keypad consists of ten number keys (0–9), the *, and the # key. The letters (a-z) are clustered into groups of three or four letters and are laid out alphabetically over keys 2 to 9. The mapping of multiple letters to a single key leads to ambiguity when a user presses a key. Various solutions for disambiguating user input exist including methods that require pressing the same key multiple times (MultiTap), consecutive presses of multiple keys (two-key disambiguation), or use of linguistic information (T9) to choose the correct letter. Wigdor *et al.* [9] introduced another method for disambiguation, chording (the concurrent presses of multiple keys), and presented a text entry method that utilize a

¹<http://www.gsmworld.com/technology/services/messaging.htm>

standard 12-key keypad *with the addition of extra keys* to provide a chording interface for entering text.

In this paper we provide an overview of some existing techniques for entering text on a mobile phone keypad. We then introduce a novel two-thumb chording method for text entry that utilizes only keys present on a standard 12-key keypad. Finally, we present a Fitts' Law based performance model of expert use. The model provides a behavioral description of the user and predicts a text entry rate of 55.02 wpm.



Figure 1: The 12-key keypad for a Motorola RAZR

2. EXISTING MOBILE PHONE TEXT ENTRY METHODS

2.1 MultiTap

MultiTap is the most common text entry method used on mobile phones. When using MultiTap the user presses a key to enter the first letter assigned to that key. Subsequent presses of the same key will cycle through the letters on that key. If the next letter of the word is on the same key, the user may wait for a timeout or press a "timeout kill" key and begin the process again. If the next letter is on a different key then the user can press that key to initiate the process again. Previous research has predicted maximum expert

words-per-minute (wpm) typing rates of 20 to 27 wpm [8]. Usability studies have found rates of 15.5 wpm [5] and 19.8 wpm [2] which are far below many other keyboard input devices. MultiTap has a key stroke per character (KSPC) value of 2.0342 [4].

2.2 Two-key Disambiguation

When using the two-key method all characters are entered with two consecutive key presses. For example, to enter the letter “s” the user first presses the 7 key (the key with the letters “pqrs”) and then the 4 key (the position of that letter on the key). While this method is simple to learn, it is not often seen outside of text entry literature. This method has a KSPC value of 2, because all letters require exactly 2 key presses to enter.

2.3 T9

When using T9, each key is pressed only once, and disambiguation is performed with the addition of linguistic information. It is possible that a series of key presses map to multiple words or that the intended word is not in the dictionary. In the former case, the most common word is displayed, and subsequent words are accessed by pressing a “next” key. In the latter case a fallback system such as MultiTap must be used to input the desired word. In either case this system introduces cognitive and procedural overhead not found in either of the above methods. Previous research has predicted expert entry rates of 41 to 46 wpm; however, these predictions assume that all words entered are unambiguous and that all words entered are in the dictionary [8]. The use of a dictionary brings T9’s KSPC value to 1.0072 when the user only enters words found in the dictionary. This assumption is not valid when text input occurs for text-messaging, where users employ a variety of abbreviations and slang [1].

2.4 ChordTap

Wigdor *et al.* introduced a chording solution to address the problems with disambiguation [9]. In their solution, three additional keys were attached to a mobile phone. By pressing one of the three chording keys in combination with a key from 2–9 the user selected the first, second, or third letter on that key. Entering “s” and “z” required pressing two of the chording keys with the 7 and 9-key respectively. After using the system for 160 minutes, divided over two days, average speeds for ChordTap reached 16.06 wpm. Curves fit to the entry rates of the participants showed that with further practice the entry rates would increase. In their experiment, ChordTap outperformed both single- and dual-handed MultiTap and showed a 90% improvement from the first session to the last. This study illustrates that chording-based systems are learnable and potentially offer an advantage over standard input systems. Unfortunately, the ChordTap user study was preliminary and does not provide entry rates for experts. The KSPC value for ChordTap is 1.84.

3. TWO-THUMB CHORDING

We now present our two-thumb chording method. Our method consists entirely of one and two-key chords using only the standard 12 keys found on mobile phone keypads. Keys 2–9 are used as *character* keys while the *, 0, # and 1 keys are used as *chording* keys. By pressing any of the character keys alone the user enters the first letter displayed on the key. Simultaneously pressing a character key in combination with the * key allows the user to enter the second letter displayed on the key, pressing a character key with the 0 key enters the third character and pressing a character key with # key enters the fourth character in the case of “s” and “z”. The space character is entered by pressing the # key alone. Table

1 shows the keymap for generating all of the letters and the space character. Numbers are entered by using the 1 key as a chording

Character Key	Chord Key				
	None	*	0	#	1
None				–	1
2	a	b	c		2
3	d	e	f		3
4	g	h	i		4
5	j	k	l		5
6	m	n	o		6
7	p	q	r	s	7
8	t	u	v		8
9	w	x	y	z	9

Table 1: Keymap for chording on a mobile phone keypad.

key to select the numeral on the key. By pressing 1 alone, the user enters 1. By pressing 1 in combination with any other number, the user enters that number. Capitalizing a letter is performed by first pressing the 0 key alone and then entering the character. Rapidly pressing 0 twice allows the user to enter caps-lock mode, pressing 0 again once reverts to the previous mode. Common symbols can be accessed by pressing the # key and character keys other than 7 or 9.

4. PREDICTIVE MODEL

Our model consists of three separate components: a Keyboard Model, a Timing Model, and a Behavioral Model. We generate our timing and keyboard models by adapting the construction techniques used by MacKenzie *et al.* to model two-thumb text entry on mini-QWERTY keyboards [6]. We then use Fitts’ Law to construct a behavioral model of an expert user employing our two-thumb chording method to input text on a mobile phone.

4.1 Keyboard Model

First we used a digitized image of a Motorola RAZR keypad as our representative keypad. We measured the keys in pixels using an image processing application. Unit of measurement does not matter as the units cancel in the Fitts’ law equation. The smaller of the width and height dimension is used as the size in accordance with prior Fitts’ Law research [3]. The keys had a width of 60 pixels. The *X* and *Y* positions of each key are shown in Table 2.

Y Pos.	X Pos.		
	50	173	292
255	1	2	3
315	4	5	6
375	7	8	9
435	*	0	#

Table 2: *X, Y* positions for a digitized Motorola RAZR keypad.

4.2 Timing Model

Our second step is to determine the functions which govern the thumb-movement and key-press time. Based on existing research [6] we introduce t_{REPEAT} as the time between repeated presses of the same key and t_{MIN} as the minimum time between key presses with alternate thumbs.

$$t_{MIN} = 1/2 \times t_{REPEAT} \quad (1)$$

Silfverberg *et al.* [8] provides a Fitts' Law model for pressing keys on a phone keypad with the thumb on the preferred hand:

$$t_{FITTS} = 176 + 64 \times \log_2(A/W + 1) \quad (2)$$

where A is the amplitude of the movement and W is the width of the target. We will use this model for both thumbs, as in MacKenzie and Buxton [3]. Additionally, the equation provides us with a value for t_{REPEAT} of 176 ms which means our tentative value for t_{MIN} is 88 ms.

4.3 Behavioral Model

Our third step is to determine a policy describing how an expert user's thumbs would move to enter characters. Given the small size of the phone keypad we will not explicitly enforce a key to thumb assignment. Instead, we will focus on a policy that examines the thumb position for the current character and determines the optimal movement of each thumb to enter the next character. We have identified four conditions that occur when inputting text with two thumbs and describe how an expert user behaves in each conditions.

1. The current character is a *single* key chord and the next character is a *single* key chord (i.e. a to d).
2. The current character is a *single* key chord and the next character is a *dual* key chord (i.e. a to e).
3. The current character is a *dual* key chord and the next character is a *single* key chord (i.e. e to a).
4. The current character is a *dual* key chord and the next character is a *dual* key chord (i.e. q to u).

4.3.1 Single to Single

In this condition the user enters two characters in a row which require the use of only one thumb. In an expert use situation, the second thumb moves to the second letter to be typed as the first thumb makes the initial keypress. The amount of time required to perform this operation will be t_{MIN} . The exception is when targeting the next character would require the unused thumb to move through the thumb that is in use. In this situation we posit that the thumb in use will finish entering the current character and then move to enter the next character. The time for this transition would be determined by t_{FITTS} .

4.3.2 Single to Dual

In this condition the user has entered a character which requires the use of only one thumb and the next character will require the use of two thumbs. Within this condition there are three possibilities.

The first possibility is that the thumb in use is already over the character key of the two key chord that will be entered. For example, when entering the word "mop" the "m" and "o" share the same character key. In this event the expert user, while entering the initial character, would move the unused thumb to the chord key needed to enter the next character. The time for this possibility is t_{REPEAT} , the time it takes to press the character key twice.

The second possibility is that the left thumb moves to the character key, and the right thumb moves to the chord key. If a thumb is not in use then its time is t_{MIN} while the time of the thumb in use is determined by t_{FITTS} . The time for this possibility is taken as the maximum of the time for each thumb.

The third possibility is that the right thumb moves to the character key, and the left thumb moves to the chord key. As in the second possibility, if a thumb is not in use then its time is t_{MIN} while the time of the thumb in use is determined by t_{FITTS} . The time for this possibility is taken as the maximum of the time for each thumb.

We select the minimum time of these three possibilities given the physical constraints from the previous key press.

4.3.3 Dual to Single

In this condition the user has entered a character which requires the use of two thumbs, and the next character will be entered with one thumb. The transition time for this task is computed as the minimum of the t_{FITTS} times for either thumb to acquire and press the key for the next character.

4.3.4 Dual to Dual

In this condition the user has entered a character which requires the use of two thumbs, and the next character also requires both thumbs. Within this condition there are three possibilities.

The first possibility is that the character key is in a column which is to the left of the column of the chord key. The time for this possibility is the maximum of the t_{FITTS} times for the left thumb to enter the character key and the right thumb to enter to the chord key. This assignment of thumbs to keys is done to prevent the crossing of thumbs. We believe that crossing the thumbs is awkward and would not accurately represent the behavior of an expert user.

The second possibility is that the character key is in a column which is to the right of the column of the chord key. The time for this possibility is the maximum of the t_{FITTS} times for the right thumb to enter the character key and the left thumb to enter the chord key. Again this assignment of thumbs to keys is done to prevent crossing of thumbs.

The third possibility is that both character and chord keys are on the same column. In this case, times for both of the previous two possibilities are computed and the time for entering the character is the minimum of the two.

5. RESULTS

The implementation of our model allows us to vary several parameters such as the slope coefficient of our Fitts' Law model, t_{MIN} , and the key width. Our corpus for testing the model is the MacKenzie and Soukoreff phrase set [7]. The corpus properly models the letter and bigram frequencies of English and is the most commonly used corpus in text entry method research. With our default parameters, our model predicts an average entry rate of 55.02 wpm for expert users. Using a standard method for calculating KSPC [4], our method has a KSPC of 1.6.

6. SENSITIVITY ANALYSIS

We perform a sensitivity analysis to explore the sensitivity of our model to changes in input parameters such as the slope coefficient of our Fitts' Law model, t_{MIN} , and the key width.

6.1 Slope Coefficient

For our timing model we use the Fitts' Law model for pressing keys on a phone keypad with the thumb as described by Silfverberg *et al.* [8]. In this model the slope coefficient is 64 ms/bit. To determine the effect of the slope coefficient on the stability of our model we systematically alter the slope coefficient and generate text entry rate predictions. Table 3 shows the predicted entry rates as we vary the slope coefficient by +/-10%, 20% and 50%.

As expected, there is an inverse relationship between slope coefficient and the predicted entry rate. This is reasonable because as the slope coefficient increases, movement time increases which should lower the predicted entry rate. Predictions stay within 15% of the nominal prediction even with variations of 50% from the nominal slope coefficient.

Slope Coefficient		Predicted WPM	
Value	% of Nominal	Value	% of Nominal
32.0	50%	62.84	114.2%
51.2	80%	57.90	105.2%
57.6	90%	56.42	102.6%
64.0*		55.02*	
70.4	110%	53.68	97.6%
76.8	120%	52.41	95.3%
96.0	150%	48.93	89.0%

Table 3: Sensitivity to slope coefficient. (* Nominal values)

6.2 t_{MIN}

As in MacKenzie and Soukoreff [6], our model assumes that the minimum time between key presses, t_{MIN} , with alternate thumbs is 88 ms. That is $\frac{1}{2}$ of the time it would take to simply press the same key repeatedly. This assumes that a user who is pressing two keys in an alternating manner begins to press down on the second key as the first key is being released, thereby doubling their entry rate. We show the effect of varying the t_{MIN} parameter on the predicted entry rate in Table 4. The influence of t_{MIN} is less than that of the slope coefficient. Changes of 50% in t_{MIN} only lead to a 4% change in the predicted entry rate.

t_{MIN}		Predicted WPM	
Value	% of Nominal	Value	% of Nominal
44.0	50%	56.67	103.0%
70.4	80%	55.66	101.2%
79.2	90%	55.34	100.6%
88.0*		55.02*	
96.8	110%	54.70	99.4%
105.6	120%	54.39	98.9%
132.0	150%	53.48	97.2%

Table 4: Sensitivity of model to t_{MIN} . (* Nominal values)

6.3 Key Width

While our Fitts' Law model uses the minimum of the key dimensions as the width, it is important to note that the width of the thumb is a confound. When using thumbs to input text, as opposed to a stylus, the effective key width may be larger. Here we show the effect of increasing key widths by 10%, 20% and 50%. The influence is minimal with a 50% increase in key width, increasing the entry rate by 3%.

Key Width		Predicted WPM	
Value	% of Nominal	Value	% of Nominal
60.0*		55.02*	
66.0	10%	55.50	100.9%
72.0	20%	55.88	101.6%
90.0	50%	56.70	103.1%

Table 5: Sensitivity of model to key width. (* Nominal values)

6.4 Behavioral Variations

Our behavioral model predicts optimal movement from one chord to the next. An alternative to this behavior is one where the user always keeps one thumb on the row of chording keys. Here we show the effect of keeping the left or right thumb on the chording row at all times.

	Predicted WPM	
Chording Thumb	Value	% of Nominal
Either	55.02*	
Left	47.88	87.02%
Right	48.49	88.13%

Table 6: Sensitivity of model to behavior. (* Nominal values)

7. CONCLUSION & FUTURE WORK

In this paper, we present a novel method and model of text entry for a standard 12-key mobile phone keypad. The model is composed of a keypad model, a Fitts' Law-based timing model, and a behavioral model of an expert user. The predicted expert text entry rate is 55.02 wpm which compares favorably to existing text entry methods. A sensitivity analysis of the model shows that it is stable under large variations to the input parameters. Future work will include a longitudinal empirical study that will train novice users until they become expert users to determine the validity of the predictions, the actions described in the behavioral model and the ability of two-thumb chording to be a viable method for inputting text on a mobile phone.

8. ACKNOWLEDGEMENTS

This material is based upon work supported, in part, by ETRI and the National Science Foundation under Grant No. 0812281. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

9. REFERENCES

- [1] R. E. Grinter and M. A. Eldridge. y do tngrs luv 2 txt msg? In *Proc. of ECSCW'01*, pages 219–238, Norwell, MA, USA, 2001. Kluwer Academic Publishers.
- [2] K. Lyons, T. Starner, D. Plaisted, J. Fusia, A. Lyons, A. Drew, and E. W. Looney. Twiddler typing: one-handed chording text entry for mobile phones. In *Proc. of CHI '04*, pages 671–678, New York, NY, USA, 2004. ACM.
- [3] I. Mackenzie and W. Buxton. Extending Fitts' law to two-dimensional tasks. In *Proc. of CHI '92*, pages 219–226. ACM Press, 1992.
- [4] I. S. MacKenzie. Kspc (keystrokes per character) as a characteristic of text entry techniques. In *Proc. of Mobile HCI '02*, pages 195–210, London, UK, 2002. Springer-Verlag.
- [5] I. S. MacKenzie, H. Kober, D. Smith, T. Jones, and E. Skepner. Letterwise: prefix-based disambiguation for mobile text input. In *Proc. of UIST '01*, pages 111–120. ACM Press, 2001.
- [6] I. S. MacKenzie and R. W. Soukoreff. A model of two-thumb text entry. In *Proc. of GI '02*, pages 117–124. Canadian Information Processing Society, 2002.
- [7] I. S. MacKenzie and R. W. Soukoreff. Phrase sets for evaluating text entry techniques. In *Proc. of CHI '03*, pages 754–755, New York, NY, USA, 2003. ACM.
- [8] M. Silfverberg, I. S. MacKenzie, and P. Korhonen. Predicting text entry speed on mobile phones. In *Proc. of CHI '00*, pages 9–16. ACM Press, 2000.
- [9] D. Wigdor and R. Balakrishnan. A comparison of consecutive and concurrent input text entry techniques for mobile phones. In *Proc. of CHI '04*, pages 81–88, New York, NY, USA, 2004. ACM.