Optimization of Split Keyboard Design for Touchscreen Devices

by

Mohammad Ali Alamdar Yazdi

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Computer Science

Auburn, Alabama
May 5, 2018

Keywords: Text Entry, Touch-typing, Mobile Devices, QWERTY Split Keyboard,
Two-Thumb Typing, Mathematical Modeling.

Approved by

Cheryl D. Seals, Chair, Associate Professor of Computer Science and Software Engineering
Fadel M. Megahed, Assistant Professor of Farmer School of Business, Miami University
Richard Chapman, Associate Professor of Computer Science and Software Engineering
Alexandar Vinel, Assistant Professor of Industrial and Systems Engineering

ABSTRACT

While many large touchscreen devices have a split keyboard option to enable two-thumb typing, there are significant differences in the design which affect users' text entry rate. A mixed integer-programming model is developed to optimize key-to-thumb assignment with the objective of minimizing the expected time to type a character. Computer simulations are conducted to determine the optimal key dimension under different values of the Fitts' law's slope coefficient, typing error rate, and alternate-hand advantage phenomenon. The results show that text entry rate and the optimal key-to-thumb assignment depend on key dimensions, user's speed-accuracy profile, and the level of alternate-hand advantage. The optimal keyboard is proposed. To validate the analytical findings, an empirical study is conducted with eighteen users and six different keyboards in terms of key dimensions and typing zone. Empirical results report between 7% and 18% improvement in the text entry rate over the other split keyboards tested.

ACKNOWLEDGMENTS

I would like to express my gratitude and thanks to my advisor, Dr. Cheryl D. Seals, for supporting my graduate studies and research. I would also like to thank my thesis committee members, Dr. Fadel M. Megahed, Dr. Richard Chapman, and Dr. Alexandar Vinel, and also Dr. Lora Cavouto for reviewing my work and providing valuable and constructive feedback. My thanks also go to Auburn University for its excellent academic service and financial support. My study at Auburn University has really been a rewarding experience. I also extend my gratitude to Dr. Ashkan Negahban who worked alongside me during the initial stages of this research. Finally, I wish to thank my family for their unconditional support. It should be noted that part of this thesis is a paper published in 2018 in the International Journal of Human-Computer Interaction (Alamdar Yazdi, Negahban, Cavuoto, & Megahed, 2018).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION AND LITERATURE REVIEW

Improving the design of touchscreen keyboards has become increasingly important as the usage of these devices continues to grow. The reduction in typing time improves the efficacy of smartphones and reduces the likelihood of negative health-related outcomes including Carpal Tunnel Syndrome (CTS) (See Marklin, Simoneau, & Monroe, 1999; Simoneau, & Marklin, 2001). The reader should note that the nature of typing on touchscreen handheld devices is significantly different than physical keyboards. For example, the assignment of keys to fingers in physical keyboards is fairly standardized through the ten-finger typing approach (Francis, & Oxtoby, 2006). On the other hand, the usage of touchscreen handheld devices varies differently by age, purpose (texting, emails, internet browsing, etc.), and device type (Gao, & Sun 2015; Smith, & Chaparro, 2015). Thus, there are unanswered research questions regarding the optimal keyboard designs/dimensions and the best assignment of keys-to-fingers (typically two thumbs) for touchscreen handheld devices.

Azenkot, & Zhai (2012) and Hsiao et al. (2009) showed that one-thumb, two-thumb are the most common typing styles on touchscreen keyboards. In this paper, we consider these two typing methods, while incorporating typing errors. It should be noted that typing errors can represent a significant workload on one thumb if it is also responsible for pressing the backspace button when a typing error occurs. The incorporation of typing errors has not been addressed in the handheld devices literature.

To facilitate two-thumb typing on large touchscreen devices, many of these products allow the keyboard to be split into two parts that appear on the sides of the screen (i.e., split keyboard). While low text entry rates are reported as one of the main disadvantages of two-thumb typing on large touchscreen devices (Oulasvirta et al., 2013), there has been little or no research on the optimal design for split keyboards in terms of key dimensions and assignment of keys to each half of the split keyboard. The goal of this paper is to optimize the QWERTY split keyboard design to improve text entry rate.

The majority of previous studies focus on changing keyboard layout. For instance, Oulasvirta et al. (2013) proposed the KALQ layout to improve text entry rate for two-thumb typing on mobile touchscreen devices by minimizing thumb travel distance and maximizing alternating between thumbs. For a list of other studies that propose alternative keyboard layouts, see Zhai, Hunter, & Smith, 2000; Zhai, Hunter & Smith, 2002. Although for many of these keyboards a higher text entry rate in terms of words per minute (WPM) by specially trained users were reported, many every-day users are accustomed to the standard QWERTY layout for the English language due to its ubiquity, and are often reluctant to learn these other keyboard layouts limiting their widespread adoption (Bi, Smith, & Zhai, 2010; Li, Guy, Yatani, & Truong, 2011; MacKenzie, Zhang, & Soukoreff, 1999). Our work differs from these studies as follows. Rather than proposing a new layout, we aim at improving text entry rate for the QWERTY split keyboard by optimizing "typing zone" and key dimensions. Typing zone is a specific distribution of keys for each of the two fingers. Figure 1 shows two different typing zones. The primary research questions explored in this paper are:

1. What are the optimal typing zones in two-thumb typing for the split QWERTY keyboard?

2. For large handheld touchscreen devices, what is the optimal key dimension in the split

QWERTY keyboard that maximizes text entry rate?

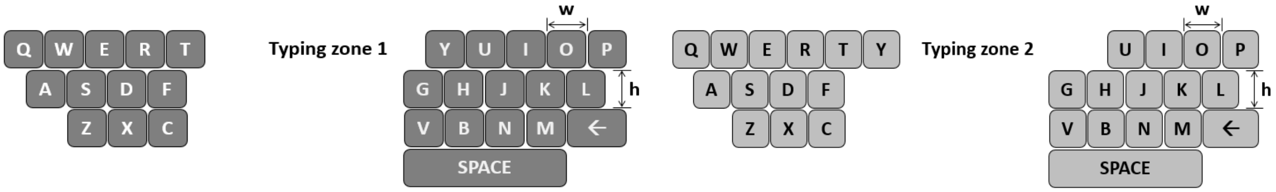


Figure 1. Two split QWERTY keyboards that determine two different typing zone for the left and right thumbs.

Therefore, we predict the following hypothesis.

H1. Our proposed optimal keyboard is faster to type than the other keyboards.

Keyboard dimensions for small touchscreen devices are constrained by screen size. High error rates are reported when keys are too small relative to the user’s fingers/thumbs, also known as the “fat finger” problem (Wigdor et al., 2007). Therefore, in general, product designers make keys as large as possible while retaining enough space for viewing the rest of the screen. Larger screens provide more freedom in designing keyboards with sufficiently keys large to minimize the “fat-finger” effect. The effect of key size on performance is evaluated in several studies. For example, Sears et al. (1993) and Colle, & Hiszem, (2004) reports longer text entry times and higher error rates for smaller square key sizes in the range of 5.7mm to 25mm. To the best of our knowledge, the effect of key dimensions for non-square keys on users’ typing speed is addressed only for conventional keyboards (Pereira et al., 2013; Pereira, Hsieh, Laroche & Rempel, 2014; Madison et al., 2015).

While there is a general consensus on the typing zone for fingers in 10-finger typing on traditional PC keyboards, the optimal typing zones in two-thumb typing on virtual keyboards had not been explored until the work of Negahban, Uludag & Yang (2013) who develop an integer programming model to find the optimal typing zone for fingers for six different typing methods commonly used by smartphone users. However, the study ignores typing errors and does not provide any analysis on the optimal key dimensions. This is limiting since typing errors represent a significant workload on the thumb operating the *Backspace* key. This paper extends and generalizes the results of Negahban et al. (2013) by incorporating the occurrence and correction of typing errors into the mathematical model. We also conduct computer simulations and perform empirical experiments to determine the optimal touchscreen QWERTY split keyboard.

The contribution of this paper is three-fold: (1) to the best of our knowledge, this is the first study to develop a mathematical model with the objective to optimize touchscreen split keyboards that incorporates typing errors; (2) it provides general guidelines about the optimal key dimensions; and, (3) it shows that the optimal typing zone varies based on key dimensions and the user's speed-accuracy profile.

CHAPTER 2 METHODOLOGY

We use Fitts' law in conjunction with the alternate-hand advantage phenomenon (Salthouse, 1986) to predict finger movement times. Next, a mixed-integer programming model is developed to determine the optimal typing zone for each thumb. Computer simulations are then conducted to determine the optimal keyboard design under various parameter settings. Finally, we examine actual user performance and validate the analytical findings using the recorded empirical data.

2.1 Finger Movement Time

Fitts' law is perhaps the most commonly used model to predict finger movement times (MacKenzie, 1992) which relates the movement time to travel distance and the size of the target key as follows:

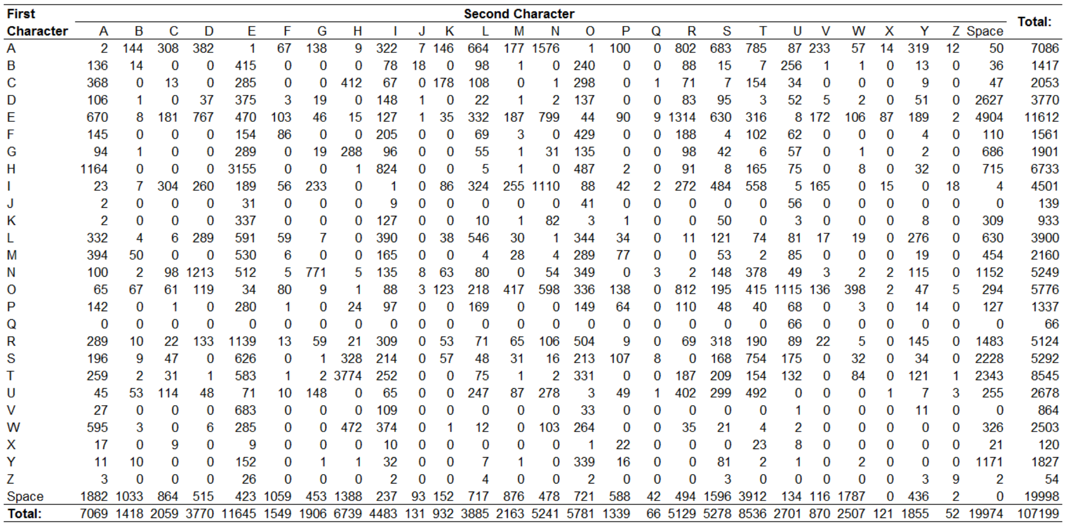
$$T_{i,j} = a + b\ log_2^{(1+\frac{D}{w})}, \tag{1}$$

where $T_{i,j}$ denotes the movement time from key i to key j, D is the physical distance between centers of the two keys, and w is the width of target key j defined as the minimum of the key's height and width (Soukoreff & MacKenzie, 1995). The intercept (a) and the slope coefficient (b) reflect the non-informational, informational aspect of pointing action, respectively (Zhai, 2004). While a pertains to the activation of muscles and key pressing actions and thus is independent of the movement distance and key size (Bi, Smith & Zhai, 2012), b reflects the time it takes to express an additional bit of information (Zhai, 2004). In the context of touch typing, coefficients a and b are commonly set to 0.083 and 0.127, respectively (Zhai et al., 2002; Bi et al., 2012). We use the same values in this paper. In the case of repeating the same character (e.g., ‘oo’ as in word “book”), we have D=0. In this case, the second term in Equation 1 is zero and the movement time will be equal to a or 0.083 seconds.

Zhai et al., (2002) propose equation (2) to calculate the expected time in seconds to type a character (t) when all keys are hit by a single pointer/stylus (equivalent to one-thumb typing in this paper).

$$t = \sum_i \sum_j Pr_{i,j} \times T_{i,j}, \tag{2}$$

where i and j correspond to any of the 26 Roman letters and the space key and $Pr_{i,j}$ represents the probability of the occurrence of digraph ij which is calculated by dividing character transition frequency for digraph ij by the sum of all character transition frequencies in the English language. The frequencies are adopted from the digraph frequency data provided in (Bi et al., 2012; Zhai et al., 2002). For completion, we present the digraph frequency data in Figure 2.



First	Second Character																										Total:
Character	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Space	
A	2	144	308	382	1	67	138	9	322	7	146	664	177	1576	1	100	0	802	683	785	87	233	57	14	319	12	50	7086
B	136	14	0	0	415	0	0	0	78	18	0	98	1	0	240	0	0	88	15	7	256	1	1	0	13	0	36	1417
C	368	0	13	0	285	0	0	412	67	0	178	108	0	1	298	0	1	71	7	154	34	0	0	0	9	0	47	2053
D	106	1	0	37	375	3	19	0	148	1	0	22	1	2	137	0	0	83	95	3	52	5	2	0	51	0	2627	3770
E	670	8	181	767	470	103	46	15	127	1	35	332	187	799	44	90	9	1314	630	316	8	172	106	87	189	2	4904	11612
F	145	0	0	0	154	86	0	0	205	0	0	69	3	0	429	0	0	188	4	102	62	0	0	0	4	0	110	1561
G	94	1	0	0	289	0	19	288	96	0	0	55	1	31	135	0	0	98	42	6	57	0	1	0	2	0	686	1901
H	1164	0	0	0	3155	0	0	1	824	0	0	5	1	0	487	2	0	91	8	165	75	0	8	0	32	0	715	6733
I	23	7	304	260	189	56	233	0	1	0	86	324	255	1110	88	42	2	272	484	558	5	165	0	15	0	18	4	4501
J	2	0	0	0	31	0	0	0	9	0	0	0	0	0	41	0	0	0	0	0	56	0	0	0	0	0	0	139
K	2	0	0	0	337	0	0	0	127	0	0	10	1	82	3	1	0	0	50	0	3	0	0	0	8	0	309	933
L	332	4	6	289	591	59	7	0	390	0	38	546	30	1	344	34	0	11	121	74	81	17	19	0	276	0	630	3900
M	394	50	0	0	530	6	0	0	165	0	0	4	28	4	289	77	0	0	53	2	85	0	0	0	19	0	454	2160
N	100	2	98	1213	512	5	771	5	135	8	63	80	0	54	349	0	3	2	148	378	49	3	2	2	115	0	1152	5249
O	65	67	61	119	34	80	9	1	88	3	123	218	417	598	336	138	0	812	195	415	1115	136	398	2	47	5	294	5776
P	142	0	1	0	280	1	0	24	97	0	0	169	0	0	149	64	0	110	48	40	68	0	3	0	14	0	127	1337
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	66	0	0	0	0	0	0	66
R	289	10	22	133	1139	13	59	21	309	0	53	71	65	106	504	9	0	69	318	190	89	22	5	0	145	0	1483	5124
S	196	9	47	0	626	0	1	328	214	0	57	48	31	16	213	107	8	0	168	754	175	0	32	0	34	0	2228	5292
T	259	2	31	1	583	1	2	3774	252	0	0	75	1	2	331	0	0	187	209	154	132	0	84	0	121	1	2343	8545
U	45	53	114	48	71	10	148	0	65	0	0	247	87	278	3	49	1	402	299	492	0	0	0	1	7	3	255	2678
V	27	0	0	0	683	0	0	0	109	0	0	0	0	0	33	0	0	0	0	0	1	0	0	0	11	0	0	864
W	595	3	0	6	285	0	0	472	374	0	1	12	0	103	264	0	0	35	21	4	2	0	0	0	0	0	326	2503
X	17	0	9	0	9	0	0	0	10	0	0	0	0	0	1	22	0	0	0	23	8	0	0	0	0	0	21	120
Y	11	10	0	0	152	0	1	1	32	0	0	7	1	0	339	16	0	0	81	2	1	0	2	0	0	0	1171	1827
Z	3	0	0	0	26	0	0	0	2	0	0	4	0	0	2	0	0	0	3	0	0	0	0	0	3	9	2	54
Space	1882	1033	864	515	423	1059	453	1388	237	93	152	717	876	478	721	588	42	494	1596	3912	134	116	1787	0	436	2	0	19998
Total:	7069	1418	2059	3770	11645	1549	1906	6739	4483	131	932	3885	2163	5241	5781	1339	66	5129	5278	8536	2701	870	2507	121	1855	52	19974	107199

Figure 2. 27*27 digraph frequency table in common English (Bi et al. 2012).

Note that Eq. (2) does not take into account typing errors. However, in touch screen devices, typing errors are common. When the target key is missed, the error is either corrected right away and before typing any additional characters (“corrected error”), or is corrected after typing additional characters or after the word/sentence is completed, or is ignored. To make the problem analytically tractable, we only account for the time to correct the "corrected errors" (hereafter referred to as typing errors or errors). The expected error correction time can be incorporated by multiplying $Pr_{i,j}$ by the error rate and then multiplying the resulting value by the movement time to and from the backspace key. There is also an *error detection time* which corresponds to a "processing" time that takes to detect a typing error. Note that the error detection time (*EDT*) is independent of movement time. Therefore, the expected processing time to detect an error is calculated by multiplying the error rate by the error detection time. Equation 3 provides the expected time for typing a character including the error correction time where *err* is the rate of occurrence of typing errors and *EDT* is the error detection time:

$$t = err * EDT + \sum_i \sum_j Pr_{i,j} \times (T_{i,j} + err \times (T_{j,Bspace} + T_{Bspace,j})), \tag{3}$$

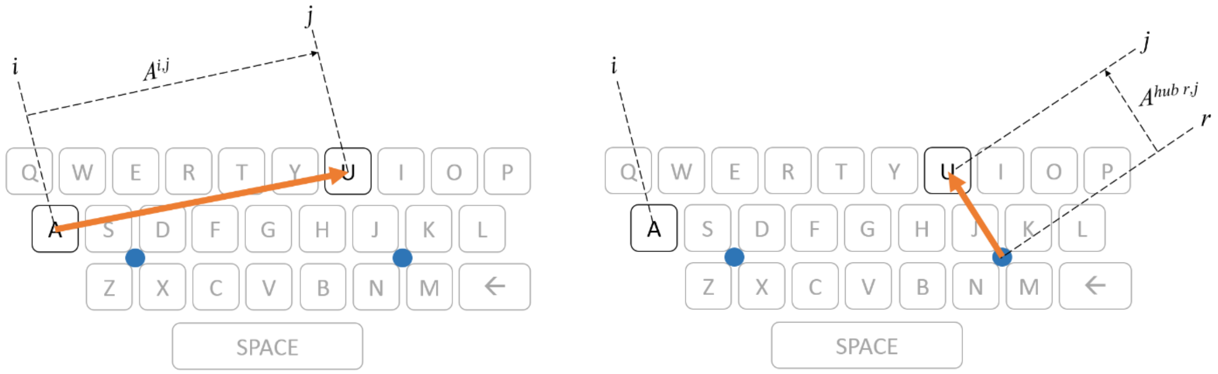
We assume the error rate is the same for all digraphs, e.g., the probability that 'a' is typed after 'c' and is missed, is assumed to be the same as when 'f' is mistyped after 'c'. Given *t* as the expected time (in seconds) to type a character, *1/t* will be the expected number of characters typed in one second and *60/t* will give the expected number of characters typed per minute. Based on the literature (Zhai et al., 2002), we consider an average of five characters per word and thus the expected number of words typed per minute (WPM) is given by

$$WPM = \frac{60/t}{5} = \frac{12}{t}. \tag{4}$$

It is worth noting that the use of word suggestion and/or auto-completion features is not considered in this model. This assumption can be justified by the findings of Quinn & Zhai (2016) who suggest that although these features can reduce the number of key strokes, they increase the average text entry time.

2.2 Modeling Two-Thumb Typing with Typing Errors

In the case of two-thumb typing, Negahban et al. (2013) determined and used the natural posture of thumbs on the touchscreen keyboard for an average user as the "hub locations" for the two thumbs (Figure 3). Figure 3(a) shows how the movement time is calculated for digraph *AU* (i.e., typing U after A is already typed) when the two letters are assigned to the same thumb while Figure 3(b) illustrates the thumb movement when letter A is assigned to the left thumb and letter U to the right thumb. To calculate the movement time, the variable *D* in Eq. (1) will correspond to the distance to key *j* from the hub location of the respective thumb (denoted by *r* in Figure 3).



(a) A and U are assigned to the same thumb

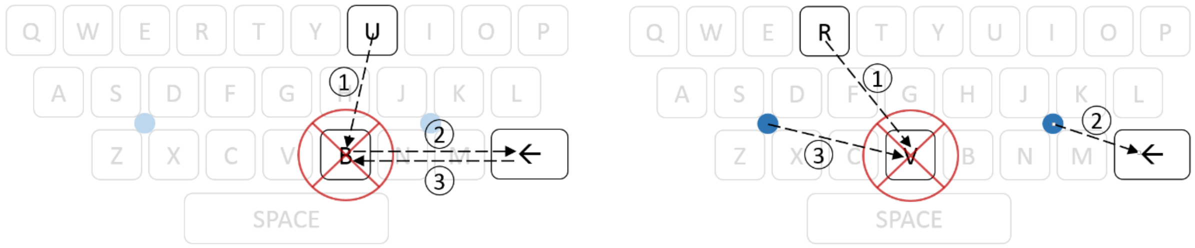
(b) A and U are assigned to different fingers

Figure 3. Hub locations (blue circles) and the two cases for movement time calculation in two-thumb typing.

Given the above, there are a number of different scenarios that need to be considered when the user misses the target character (i.e., typing error). For the sake of illustration, suppose the backspace key is assigned to the right thumb. The four possible error correction processes are illustrated in Figure 4. For the sake of conciseness, we only describe the first two cases here while the other two cases can be explained by the same analogy.

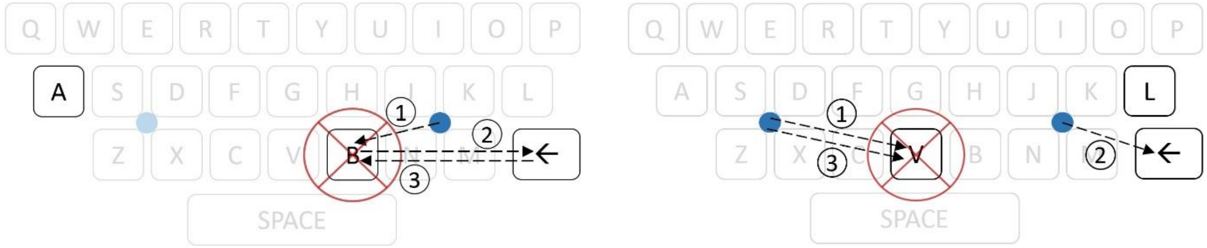
Case 1: i and *j* are assigned to the right thumb: In this case, the right thumb moves from *i* to a location around *j* (the target is missed) and from that location to the backspace key to delete the mistyped character. Assuming any point around *j* is equally likely to be hit by mistake, it would be reasonable to use the center of key *j* to represent the "average" position of the location hit by mistake. Finally, we complete the digraph by moving from the Backspace key to the center of *j*. An example of this case is shown for the digraph *UB* in figure 4(a) where both letters *U* and *B* are assigned to the right thumb. It is worth noting that this scenario applies to all digraphs *ij* in one-thumb typing since all keys are assigned to the same thumb.

Case 2: i and *j* are assigned to the left thumb: In this case, the left thumb moves from *i* to a location around *j*. Again, we will use the center of key *j* to represent the average position of this random location. Since *j* is missed and based on our assumption on the use of hub locations, the right thumb needs to move from its hub location to the center of the backspace key to delete the mistyped character. To complete the digraph, the left thumb will then move from its hub location to the center of key *j*. An example of this case for diagraph *RV* is shown in Figure 4(b).



(a) Case 1

(b) Case 2



(c) Case 3

(d) Case 4

Figure 4. Four cases of error correction in two-thumb typing (the Backspace key is assigned to the right thumb)

Finally, numerous empirical studies show that, for typists of average typing skills, successive keystrokes on alternate hands (switching between left and right hands) are 30 to 60 milliseconds faster than when the preceding keystroke was on the same hand – a phenomenon also known as the alternate-hand advantage (see Salthouse (1986) for a list of studies that confirm this phenomenon). We incorporate this typing phenomenon into the proposed mathematical model by subtracting the magnitude of the alternate-hand advantage (denoted by v) from the finger movement time for digraphs where switching occurs. For example, in cases 2 and 4, we switch hands two times leading to 2*v milliseconds faster keystrokes.

2.3 The Optimization Model

The notations of the proposed mixed-integer programming model for optimization of typing zone in two-thumb typing are summarized in Table 1.

Table 1. The notations for the mixed-integer programming model.

Parameters	Description
F	Set of thumbs containing the indices for thumbs (left and right thumbs)
L	Set of keys containing the indices for characters
i, j	Character indices; $1 \leq i, j \leq 28$ with 28 being the index number for the backspace key
l, r	Thumb indices; $1 \leq l, r \leq 2$
$Pr_{i,j}$	Probability of transition from character *i* to character *j*
$T_{i,j}$	Movement time based on Fitts' law (Eq.1) from key *i* to key *j*
$T_{j,Bspace}$	Movement time based on Fitts' law (Eq.1) from key *j* to the backspace key
$T_{Bspace,j}$	Movement time based on Fitts' law (Eq.1) from the backspace key to key *j**
$T_{l,Bspace}^{hub}$	Movement time based on Fitts' law (Eq.1) from the hub location of thumb *l* to the backspace key
$T_{l,j}^{hub}$	Movement time based on Fitts' law (Eq.1) from the hub location of thumb *l* to key *j*
err	Error rate
v	Alternate-hand advantage value
M	Large positive number (big-M)
Decision variables	
$t_{i,j}$	Movement time (cost) of typing character *j* after *i* while incorporating typing errors
$Z_{i,j}$	Is equal to 1 if character *i* is assigned to thumb *l* and 0 otherwise.

* Since the dimensions of the backspace key is different, $T_{Bspace,j}$ and $T_{j,Bspace}$ may not necessarily be equal even though the distance *D* in both cases is the same (see W in Eq. 1).

Since the amplitude and width are measured in the same units and form a ratio in Fitts' law, the movement time will not change as long as the same relative scale is maintained. Therefore, instead of the actual key dimensions, we focus on the ratio of keys' height to width (*h/w)* to calculate finger movement times. We set the width of the Space and the Backspace keys to be 1.5 and 4 times larger than the other keys, respectively, as shown in Figure 1. The model assigns every key to either the right or left thumb in order to fulfill the constraints. For example, if $Z_{7,1} = 1$ then the seventh letter of the alphabet, 'g', is assigned to left thumb, and if $Z_{7,2} = 1$, then 'g' is assigned to the right thumb. The model minimizes the expected time to type a character by finding two sets of Z values each set representing the optimal typing zone for the corresponding thumb.

The proposed mixed integer programming model is as follows:

$$\text{Min } t = \sum_{i\in L}\sum_{j\in L} Pr_{i,j} \times t_{i,j} \tag{5}$$

subject to

$$t_{i,j} \geq\ T_{i,j} + err \times \left(T_{j,Bspace} + T_{Bspace,j}\right) + M\left(Z_{i,l} + Z_{j,l} + Z_{28,l} - 3\right)$$

$$\forall i,j \in L - \{28\}\ , \forall\ l \in F, \tag{6}$$

$$t_{i,j} \geq\ T_{i,j} + err \times \left(T^{hub}_{r,Bspace} + T^{hub}_{l,j} - 2v\right) + M\left(Z_{i,l} + Z_{j,l} + Z_{28,r} - 3\right)$$

$$\forall i,j \in L - \{28\},\ \forall\ l,r \in F, l \neq r, \tag{7}$$

$$t_{i,j} \geq\ T^{hub}_{l,j} - v + err \times \left(T_{j,Bspace} + T_{Bspace,j}\right) + M\left(Z_{i,r} + Z_{j,l} + Z_{28,l} - 3\right)$$

$$\forall i,j \in L - \{28\}, i \neq j,\ \forall\ l,r \in F, l \neq r, \tag{8}$$

$$t_{i,j} \geq\ T^{hub}_{l,j} - v + err \times \left(T^{hub}_{r,Bspace} + T^{hub}_{l,j} - 2v\right) + M\left(Z_{i,r} + Z_{j,l} + Z_{28,l} - 3\right)$$

$$\forall i,j \in L - \{28\}, i \neq j,\ \forall\ l,r \in F, l \neq r, \tag{9}$$

$$\sum_{l\in F} Z_{i,l} = 1 \qquad \forall i \in L, \tag{10}$$

$$t_{i,j} \geq 0 \qquad \forall i,j \in L, \tag{11}$$

$$Z_{i,l} \in \{0,1\} \qquad \forall i \in L, \quad \forall l \in F. \tag{12}$$

The objective function (Eq. 5) is the expected value of time to type a character (t). Note that the inclusion of the error detection time is equivalent to adding a constant to the objective function and thus does not influence the optimal solution. As a result, the error detection time is removed from the optimization model. Constraints (6-9) determine the movement time to type character *j* after *i* based on the assignment of keys *i*, *j*, and *Backspace* to the two thumbs. Constraint (6) pertains to the case where keys *i* and *j* and Backspace are assigned to the same thumb that also operates the backspace key. The last expression guarantees that this cost is used only if the above condition is true. In this case, if an error occurs, the correction time will be similar to *Case 1* in Figure 4(a). If key *i*, key *j*, and *Backspace* are not operated by the same thumb, then either $Z_{i,l}$, $Z_{j,l}$ or $Z_{28,l}$ will be zero making $Z_{i,l} + Z_{j,l} + Z_{28,l} - 3$ a negative number (note that the index for the backspace key is 28). *M* is chosen to be a sufficiently large positive number and thus this would result in a large negative value for the cost. Since all constraints are expressed in the form of inequality constraints of type (≥), the model will be forced to choose the proper cost value (i.e., the only $t_{i,j}$ that is positive).

Constraint (7) pertains to the case where *i* and *j* are assigned to the same thumb while *Backspace* is assigned to the other thumb similar to *Case 2* in Figure 4(b). Once again, the last expression ensures that this constraint is binding only if the above condition is true. Since we switch between the two thumbs twice, we make use of the alternate-hand advantage phenomenon two times. By a similar analogy, Constraint (8) considers the case where keys *i* and *j* are assigned to different thumbs while *Backspace* is assigned to the same thumb as key *j* (*Case 3* as in Figure 4(c)). In constraint (9), keys *i* and *j* are assigned to different thumbs while key *i* and *Backspace* are operated by the same thumb (Case 4 as in Figure 4(d)). Constraint (10) ensures each character *i* is assigned to one and only one thumb. Constraint (11) indicates that movement time ($t_{i,j}$) is non-negative. Finally, the integrality constraint (12) declares that the decision variables pertaining to key-to-thumb assignments ($Z_{i,l}$) are all of the type binary.

In the following section, we run the model for different values of error rate and *h/w* ratio and analyze the optimal typing zone and the corresponding WPM measure.

CHAPTER 3 COMPUTER SIMULATIONS AND RESULTS

The model is solved for various key dimensions and error rates to find not only the optimal typing zone for two-thumb typing on the QWERTY keyboard but also the optimal keyboard design. Various scenarios for computer simulations are summarized in Table 2. The model is solved for the resulting 48 scenarios using the CPLEX Optimization Studio v12.4 on a typical PC with 2.5GHz dual-core CPU and 6GB of RAM (computational time for solving each problem is within a few seconds).

Table 2. Scenarios for computer simulations.

Parameter	Notation	Value/Range	Increment	Number of levels
Key height/width	*h/w*	[0.6, 2.0]	0.2	8
Error rate	*Err*	[0, 0.10]	0.02	6

3.1 Optimal Typing Zone and Keyboard Design

Figure 5 illustrates three typing zones and the corresponding key dimension and error rate for which those typing zones are found to be optimal. Perhaps the most important finding is that the optimal typing zone varies based on key dimensions and error rate, i.e., there is no single typing zone that is optimal for all keyboards and users. As the ratio of the key height to key width increases, the model generally assigns more workload to the left thumb as we see a shift in the optimality from typing zone 1 to typing zone 3. For *h/w*=1.2, two typing zones can be optimal based on the user's error rate so that typing zone 2 is optimal for *err* = 0% and 2% while typing zone 3 is optimal for *err* = 4%, 6%, 8%, and 10% and thus a shift from typing zone 2 to 3 as the error rate increases. Note that a higher error rate essentially means more usage of the Backspace key and thus higher workload for the right thumb. Due to this increased workload, the model assigns key *G* to the left thumb to get the most benefit from the alternate-hand advantage.

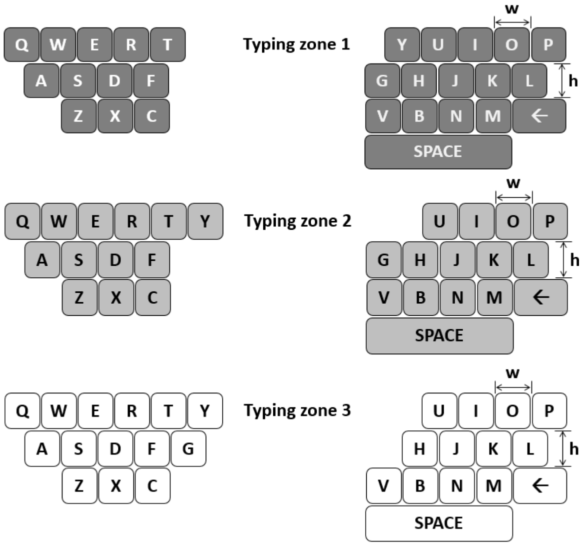


Figure 5(a) Three different typing zones found to be optimal under different parameter settings.

	Error rate					
h/w ratio	0%	2%	4%	6%	8%	10%
0.6	1	1	1	1	1	1
0.8	1	1	1	1	1	1
1.0	2	2	2	2	2	2
1.2	2	2	3	3	3	3
1.4	3	3	3	3	3	3
1.6	3	3	3	3	3	3
1.8	3	3	3	3	3	3
2.0	3	3	3	3	3	3

Figure 4(b) Optimal typing zone for different keyboard dimensions and error rates. The value of each cell shows the optimal typing zone (Figure 4(a)) for the corresponding ratio of keys' height to width (row value) and error rate (column value).

Figure 5. Optimal typing zone for different key dimensions and error rates (*v*=30 ms)

Interestingly, in all three typing zones, the space bar is assigned to the right thumb with the difference being related to characters *Y* and *G*. The difference between the three typing zones in terms of the workload on each thumb and use of the alternate-hand advantage phenomenon is summarized in Table 3. The table provides thumb transition probabilities and total workload for each thumb. For instance, the "Right thumb after left thumb" column provides the likelihood of the right thumb being activated after a character is typed by the left thumb. Therefore, the "Right thumb after left thumb" and "Left thumb after right thumb" columns represent how much the respective typing zone makes use of the alternate-hand advantage. The last two columns simply provide the total workload for the right and left thumbs, respectively.

Table 3. Thumb transition probabilities and workload of each typing zone without backspace

Typing Zone	Right thumb after right thumb	Left thumb after left thumb	Left thumb after right thumb	Right thumb after left thumb	Right thumb load	Left thumb load
1	26.3%	15.5%	29.1%	29.1%	55.4%	44.6%
2	24.0%	16.5%	29.7%	29.8%	53.7%	46.3%
3	21.2%	17.3%	30.7%	30.8%	51.9%	48.1%

The WPM measure predicted by the model for the optimal typing zone given various h/w ratios are shown in Figure 6. According to the figure, when *v=30* milliseconds, the maximum WPM occurs at *h/w=1.0*. Note that this finding is consistent for other *err* and *v* values suggesting that a keyboard with square keys (*h=w*) is optimal regardless of the user's error rate and the level of alternate-hand advantage. By analyzing the slope of the line segments in Figure 6, another important observation is that the sensitivity of the WPM measure to the h/w ratio decreases as the ratio increases. As similar patterns are observed for other error rates, it can be concluded that, for an alternate-hand advantage of 30 milliseconds and regardless of the user's error rate, the optimal design for a split QWERTY keyboard for large handheld devices such as tablets, is given by typing zone 2 and has square keys, as shown in Figure 7.

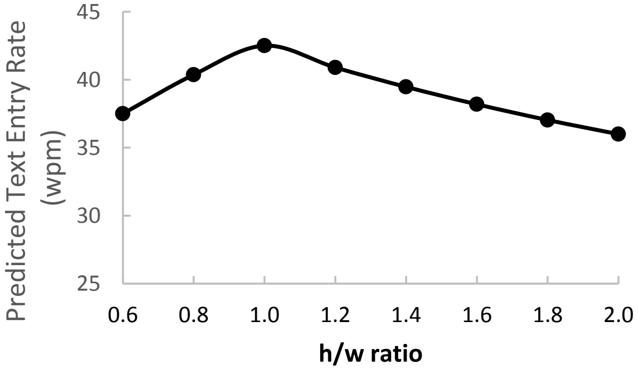


Figure 6. Predicted WPM for err=0.04, *v*=0.03, and using the corresponding optimal typing zone

Figure 7. Optimal keyboard design for alternate-hand advantage of 30 ms

It is important to note that the above results pertain to $v=30$ ms and a user whose typing personal speed-accuracy profile matches those assumed by the Fitts' law. In the next section, we analyze the effect of these two components in more detail.

3.2 The Effect of User's Speed-Accuracy Profile and Alternate-Hand Advantage

Here, we explore the effect of Fitts' law's slope coefficient (b) as well as the level of alternate-hand advantage (v) on the optimal split keyboard design.

a) Slope Coefficient. In this set of scenarios for computer simulations, the slope coefficient is perturbed by -40%, -20%, +20% and +40%. The results are summarized in Table 4 and suggest that the optimal typing zone (i.e., split keyboard) depends on the user's speed-accuracy profile. Table 5 summarizes the change in the WPM performance measure for typing zone 2 under a 4% error rate. The results indicate an inverse relationship between the slope coefficient and WPM predictions. It is worth noting that a similar sensitivity analysis on Fitts' law's slope coefficient has been performed by MacKenzie and Soukoreff (2002). Their theoretical results suggest that +20% and -20% change in coefficient slope results in -7.6% and +9% change in text entry rate whereas our numerical results indicate -12% and +16% change in text entry rate, respectively. The difference in the results stems from the modeling approach for predicting movement time and that their approach ignores typing errors (i.e, *Backspace* key) and the alternate-hand advantage phenomenon.

Table 4. The optimal typing zone broken down by the slope coefficient and error rate (square keys and *v*=0.03).

Slope Coefficient (ms/bit)		Error					
Value	% of Nominal	0%	2%	4%	6%	8%	10%
0.076	60%	2	2	2	2	3	1
0.102	80%	2	2	2	2	2	3
0.127*	-	2	2	2	2	2	2
0.152	120%	2	2	2	2	2	2
0.178	140%	2	2	2	2	2	2

* Nominal value

Table 5. WPM performance broken down by the slope coefficient (square keys, typing zone 2, *err*=0.04, and *v*=0.03)

Slope Coefficient (ms/bit)		WPM Prediction	
Value	% of Nominal	Value	% of Nominal
0.076	60%	59.1	139%
0.102	80%	49.4	116%
0.127*	-	42.5*	-
0.152	120%	37.3	88%
0.178	140%	33.2	78%

* Nominal value

b) Alternate-hand Advantage. We vary the effect of the alternate-hand advantage from 30 to 60 milliseconds. According to Table 6, as the effect increases, we observe a shift in the optimal typing zone from zone 2 to 3 which can be justified by the larger number of switching between the two thumbs in typing zone 3 (as suggested in Table 3) leading to more benefit from the alternate-hand advantage. An increase in the alternate-hand advantage is also expected to improve WMP which is in fact supported by the results (Table 7).

Table 6. Optimal typing zone broken down by the alternate-hand advantage (square keys, *b*=0.127)

Alternate Hand Advantage (ms)	Error rate					
	0%	2%	4%	6%	8%	10%
30*	2	2	2	2	2	2
40	2	3	3	3	3	3
50	3	3	3	3	3	3
60	3	3	3	3	3	3

* Nominal values

Table 7. The sensitivity of the WPM to the alternate-hand advantage for square keys, *err*=0.04

Alternate Hand Advantage (ms)	WPM Prediction	
	Value	% of Nominal
30*	45.9*	-
40	46.9	102.3%
50	48.1	104.8%
60	49.3	107.5%

* Nominal values

CHAPTER 4 WEB APPLICATION

In order to validate the proposed mathematical model and our analytical findings regarding the optimal QWERTY split keyboard, it is necessary to explore users' text entry rate under various split keyboard designs. A web application is designed and developed to provide users with different split keyboard designs in order to evaluate their text entry rates. This chapter describes this web application.

4.1 Design Requirements

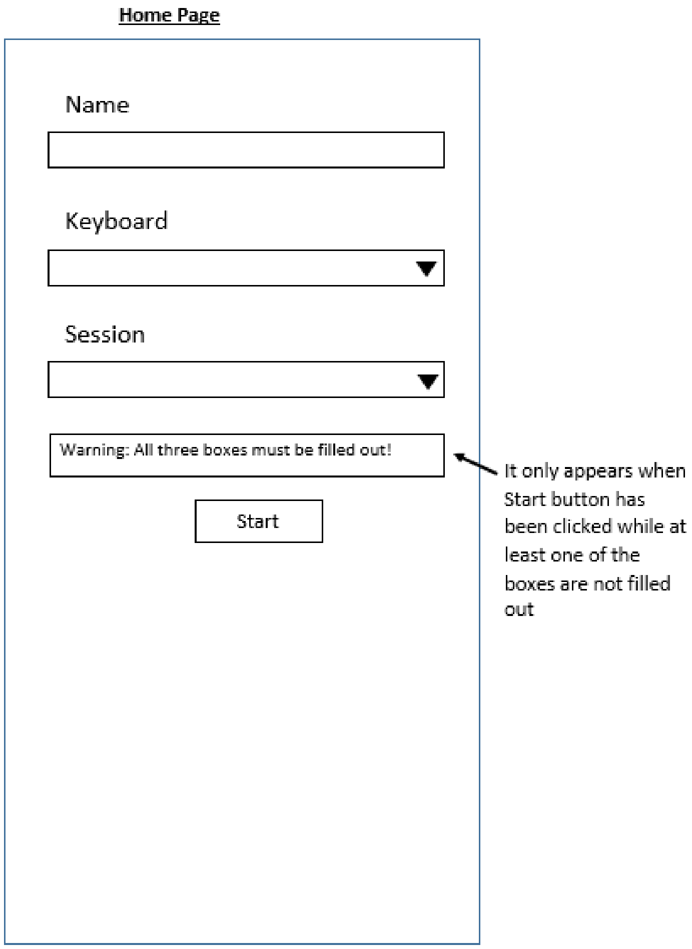
The necessary requirements of the web application are as follows:

- The design of this application should be made simple.
- This application must ask users to input their names.
- This application should create split keyboards with different designs.
- All keys in the split keyboards should be easily reachable by thumbs.
- This application should simulate the visual feedback by highlighting and showing the character above the tapped key.
- This application should ask users to type a few sentences.
- The words which users need to type and the words which have been typed so far by users should be shown right above the split keyboard.
- The split keyboards should be designed in a way that retains enough space for viewing the rest of the screen.

- There must be no delay between the time that a key is tapped by users and the visual feedback and the letter being shown on the screen.
- The split keyboards should work for different screen sizes and also for both Landscape and Portrait orientations.
- This application should be able to record required information regarding all the touch events that happen during the experiment. The collected information should be able to answer these questions:
    - What letter has been typed?
    - What letter did the user intend to type?
    - What location on the screen was touched?
    - When did the touch occur?

4.2 Wire Frame

We found it useful to build a wire frame before implementing the application. Figure 8 shows the wire frame for the home page. In this page, users type their names and select one of the options from the keyboard dropdown and one from the session drop-down before clicking “Start” button. The Start button will open the “Experiment” page if all the Name, Keyboard, and Session have been filled out; otherwise, a warning message will appear.



It only appears when Start button has been clicked while at least one of the boxes are not filled out

Figure 8. Wire frame for the homepage

Figure 9 shows the wire frame for the Experiment page. A phrase of four to five words will appear on the top of the page. Users should type each line using the split keyboards located in the bottom of the page. When all the words are typed, the page will automatically show another phrase until all the phrases in a session are typed by the user. After the end of a session, a CSV file containing user's text entry performance will automatically be downloaded on the user's device and the Home page will be loaded again for the next sessions.

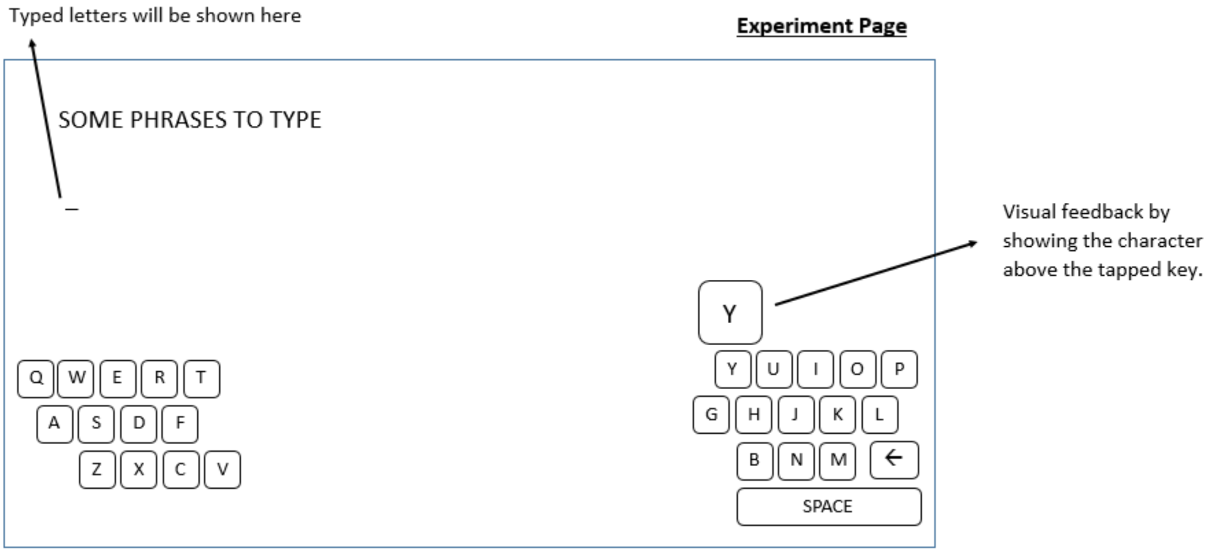


Figure 9. Wire frame of ‘experiment’ page

4.3 Software and Hardware Requirements

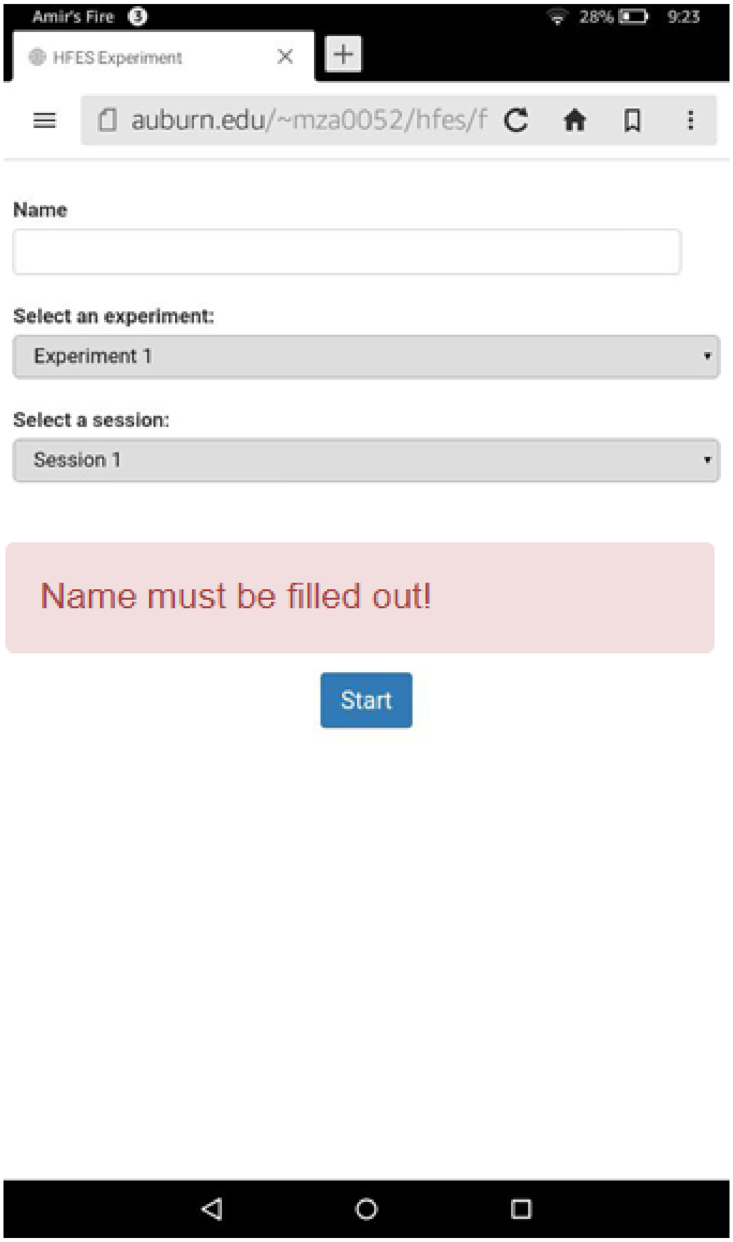
Software and Hardware Requirements of this web application are as follows:

- Hardware
  - Large touchscreen devices such as Tablets and iPads
  - Internet Connectivity
- Software
  - User interface development will use HTML, CSS, Bootstrap, and JavaScript.

4.4 Implementation

The web application was implemented in HTML, CSS, and JavaScript. In order to make the web application adaptable to various screen sizes, the developers used the Bootstrap libraries. This web application is available to the individuals with tablets, iPads or any large touchscreen devices provided that their devices have a modern web browser.

Figures 10 and 11 show the application homepage and experiment page.



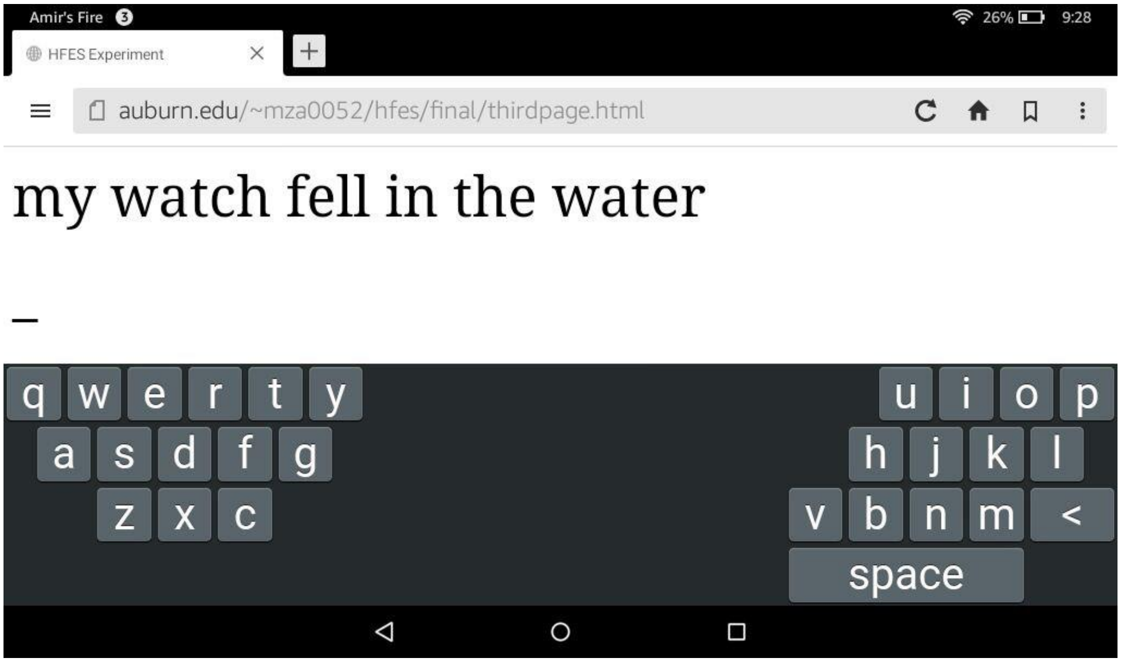
Name

Select an experiment:
Experiment 1

Select a session:
Session 1

Name must be filled out!

Figure 10. The web application homepage

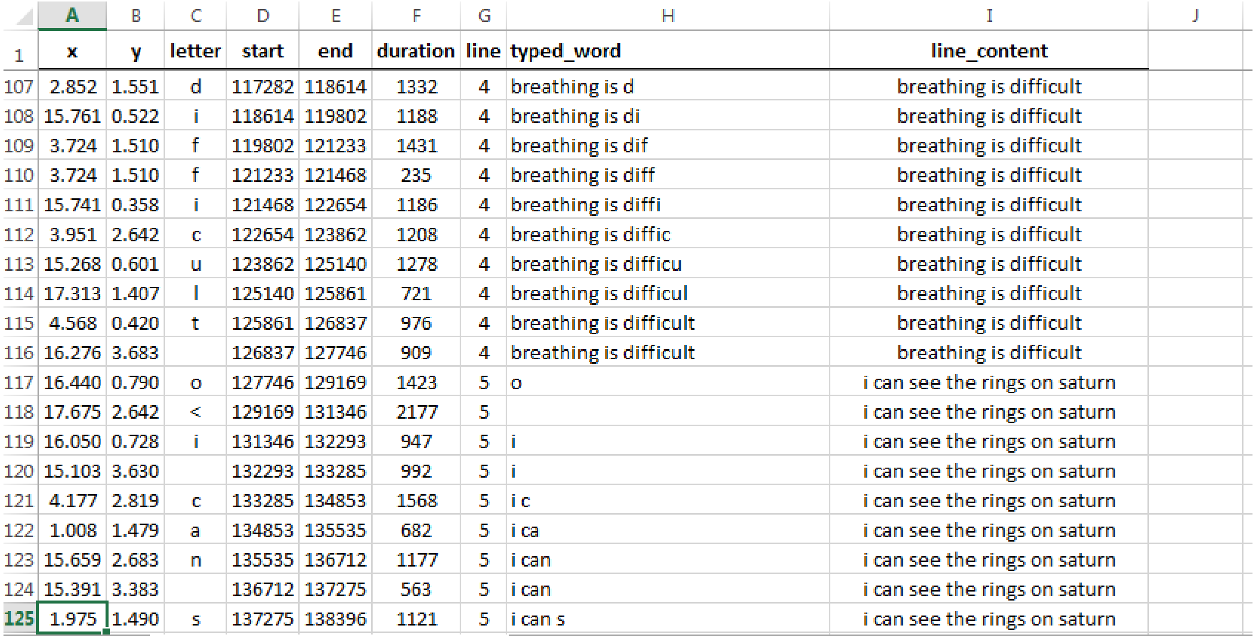


my watch fell in the water

_

Figure 11. The web application experiment page

Figure 12 shows a few lines of the output file. Each row of the data is related to one key tap. For example, the first line of figure 12 shows that it took 1332 milliseconds for the user to tap key “d”.



	A	B	C	D	E	F	G	H	I	J
1	x	y	letter	start	end	duration	line	typed_word	line_content	
107	2.852	1.551	d	117282	118614	1332	4	breathing is d	breathing is difficult	
108	15.761	0.522	i	118614	119802	1188	4	breathing is di	breathing is difficult	
109	3.724	1.510	f	119802	121233	1431	4	breathing is dif	breathing is difficult	
110	3.724	1.510	f	121233	121468	235	4	breathing is diff	breathing is difficult	
111	15.741	0.358	i	121468	122654	1186	4	breathing is diffi	breathing is difficult	
112	3.951	2.642	c	122654	123862	1208	4	breathing is diffic	breathing is difficult	
113	15.268	0.601	u	123862	125140	1278	4	breathing is difficu	breathing is difficult	
114	17.313	1.407	l	125140	125861	721	4	breathing is difficul	breathing is difficult	
115	4.568	0.420	t	125861	126837	976	4	breathing is difficult	breathing is difficult	
116	16.276	3.683		126837	127746	909	4	breathing is difficult	breathing is difficult	
117	16.440	0.790	o	127746	129169	1423	5	o	i can see the rings on saturn	
118	17.675	2.642	<	129169	131346	2177	5		i can see the rings on saturn	
119	16.050	0.728	i	131346	132293	947	5	i	i can see the rings on saturn	
120	15.103	3.630		132293	133285	992	5	i	i can see the rings on saturn	
121	4.177	2.819	c	133285	134853	1568	5	i c	i can see the rings on saturn	
122	1.008	1.479	a	134853	135535	682	5	i ca	i can see the rings on saturn	
123	15.659	2.683	n	135535	136712	1177	5	i can	i can see the rings on saturn	
124	15.391	3.383		136712	137275	563	5	i can	i can see the rings on saturn	
125	1.975	1.490	s	137275	138396	1121	5	i can s	i can see the rings on saturn	

Figure 12. A snapshot of application's output as a CSV file.

4.5 JavaScript Unit Testing

A lot of test cases were designed to ensure that the behavior of the code is the same as what we expect. The test cases were written in a separate JavaScript file to evaluate the performance of the JavaScript functions in the experiment page. For example, some tests were developed to make sure that the tapped key is the same as the typed letter on the screen. Another example is a set of tests related to the accuracy of the experiment output.

CHAPTER 5 VALIDATION STUDY

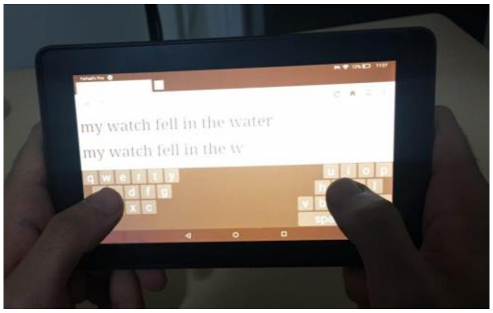
We examined users' typing performance on six different split keyboards. Participants with no prior experience with split keyboards were selected to decrease bias and the effect of experience on performance.

5.1 Participants

Eighteen graduate level students (12 Male, 6 Female) with an average age of 29 years (Std. Dev. = 3 years) volunteered to type multiple phrases (in multiple sessions) on six different touchscreen split keyboards in terms of key dimension and/or typing zone. The experiment was approved by the University at Buffalo Institutional Review Board. Informed consent was obtained from each participant. Participants had no visual impairments or typing-related disabilities, had experience with touchscreen QWERTY keyboards but no prior experience with split keyboards. All eighteen participants were right-handed.

5.2 Apparatus

The experiment involved typing short sentences on a Kindle Fire® 7" tablet (with no protection cover case). Participants were asked to hold the tablet (in the horizontal orientation) with both hands and type sentences by two-thumb typing. We evaluated six different split keyboards using a web app developed by the authors (Figure 13). Word suggestion, auto-completion and auto-correction features were not incorporated into the app. The app simulated the Android keyboard's visual feedback by highlighting and showing the character above the tapped key. Holding down a key has the same effect as a single tap. For example, in some keyboards holding backspace will result in deletion of multiple letters depending on the length it is held, but in this keyboard, no matter how long the backspace button is held, it will only delete one character.



(a) A participant typing with the app


my watch fell in the water

my watch fell in the water

(b) A screenshot of Keyboard 1 (c) A screenshot of Keyboard 2


my watch fell in the water

my watch fell in the water

(d) A screenshot of Keyboard 3 (e) A screenshot of Keyboard 4


my watch fell in the water

my watch fell in the water

(f) A screenshot of Keyboard 5 (g) A screenshot of Keyboard 6

Figure 13. The web application.

5.3 Study Procedure

Each subject completed ten sessions on each of the six keyboards. The presentation order for the keyboard designs was based on a balanced Latin Square design of experiment. Each session involved typing five phrases (between 24 to 31 words), with a 30-second break after each session. Phrases were randomly chosen from a pool of 500 phrases proposed by MacKenzie & Soukoreff (2003). The subjects were given a five minute break after finishing the ten sessions for any split keyboard. Participants were instructed to type "as accurately and as naturally" as possible. Backspace key could be used to delete the incorrect characters. The experiments were performed in an office environment.

WPM for each subject and for each session were collected. Word count was defined by the number of characters typed including the *space* divided by 5 (the first character of each phrase was removed from the WPM calculation). WPM was then obtained by dividing the word count by the typing time (in minutes). An average value of the WPM values for the 10 sessions for each subject and for each keyboard were calculated and used in the analysis.

Table 8 present the experimental design. The keyboards differ by the typing zone and key dimensions. Keyboards 1, 2 and 3 pertain to typing zones 1, 2, and 3, respectively, as in Figure 5 and Figure 13(b-d) while Keyboard 6 pertains to the typing zone as shown in Figure 13(g). The reason behind the choice of this typing zone was to simply evaluate an additional alternative that is close to the other three but was not found, in our computer simulations, to be potentially optimal. Keyboards 1, 2, 3, and 6 have a *h/w* ratio of 1.0. The reason for this choice is that the model determined this ratio to be optimal. We hypothesized that Keyboard 2 would result in the best performance and Keyboard 6 the worst because its typing zone is different than the three possible optimal typing zones identified by the mathematical model. Keyboards 2, 4, and 5 have different *h/w* ratio but they all use typing zone 2, as in Figure 13(c, e-f). We hypothesized that the performance of Keyboard 2 would be better than Keyboard 5, and Keyboard 5 would outperform Keyboard 4 (based on previous results).

Table 8. The experimental design for the validation study

Keyboard	h/w ratio	Typing zone	w(mm)	h(mm)
Keyboard 1	1.0	1	8.0	8.0
Keyboard 2*	1.0	2	8.0	8.0
Keyboard 3	1.0	3	8.0	8.0
Keyboard 4	0.8	2	8.5	6.8
Keyboard 5	1.25	2	7.4	9.2
Keyboard 6	1.0	4	8.0	8.0

* The optimal keyboard based on our numerical experiment with the mathematical model

The values for key height and width for each given *h/w* were selected such that each half of the keyboard is within the reach of the corresponding thumb (Figure 14). Oulasvirta et al. (2013) reported that the smallest thumb reach among different two-hand grips is 57.6 mm. Therefore, we considered a thumb sweep radius of 57.6 mm as the active region in designing the split keyboards.

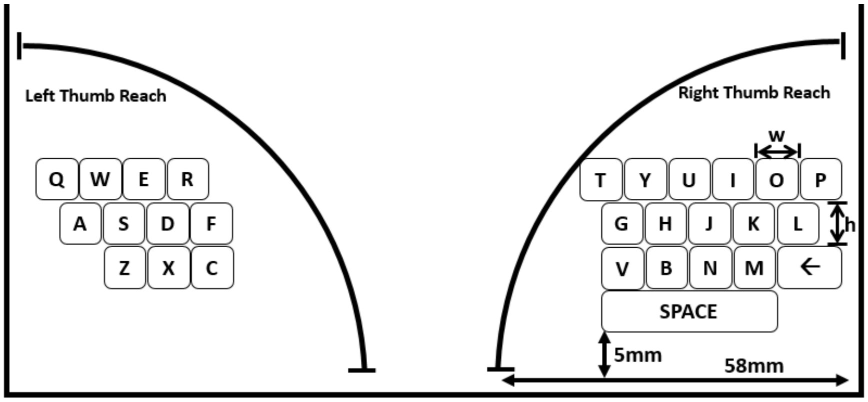


Figure 14. Width (*w*) and height (*h*) of keys are chosen so that each side of the split keyboard will be inside the thumb sweep radius of 57.6 mm.

5.4 Statistical Tests

A within-subjects analysis of variance (ANOVA) was used to test whether the six keyboards have different text entry rates. The level of statistical significance is set at $p < .05$ in all of the following analysis and the mean values are presented with their corresponding 95% confidence interval. If a significant difference is found, post-hoc comparisons using LSD are performed.

5.5 The Results of the Empirical Study

A significant difference was found between the text entry rate (WPM) of the six keyboards ($F_{5,85}$ = 10.51, *MSE* = 46.28, $p < .001$). About 38.2% of the variance in text entry rate was explained by keyboard type ($\eta^2 = .382$). We used LSD pairwise comparisons to determine how text entry rate differed across the six keyboards. Post hoc analysis showed that the performance on Keyboard 2 (our proposed optimal split keyboard) was statistically superior to Keyboard 3 ($p = .022$), Keyboard 4 (p < .001), Keyboard 5 ($p = .013$), and Keyboard 6 ($p < .001$). There was no significant difference between text entry rate of Keyboard 2 and Keyboard 1 ($p = .087$). See Table 9 for descriptive statistics by Keyboard type. Therefore, the empirical results confirm our hypotheses and validate the proposed mathematical model and analytical findings on the optimality of the keyboard with square keys and typing zone 2.

Table 9. Performance (WPM) of the six keyboards in our validation study

Keyboard	Average WPM	Std. Dev. (WPM)	N	% of Optimal
Keyboard 1	26.4	5.7	18	96%
Keyboard 2*	27.6	6.0	18	-
Keyboard 3	26.0	5.7	18	94%
Keyboard 4	23.3	4.4	18	84%
Keyboard 5	26.2	5.5	18	95%
Keyboard 6	23.9	5.2	18	86%

*The theoretical optimal keyboard

Figure 15 presents the observed WPM value for typing zones 1, 2, 3 and 4 under square keys (i.e., Keyboards 1, 2, 3 and 6, respectively). The post-hoc analysis showed that typing zone 4 is significantly slower than the other three ($p = .012$ for Keyboards 1, $p < .001$ for Keyboard 2, and $p = .040$ for Keyboard 3). This is consistent with our model's prediction since typing zone 4 (Keyboard 6) is not among the three possible optimal zones (Keyboards 1, 2, and 3) as identified by the model.

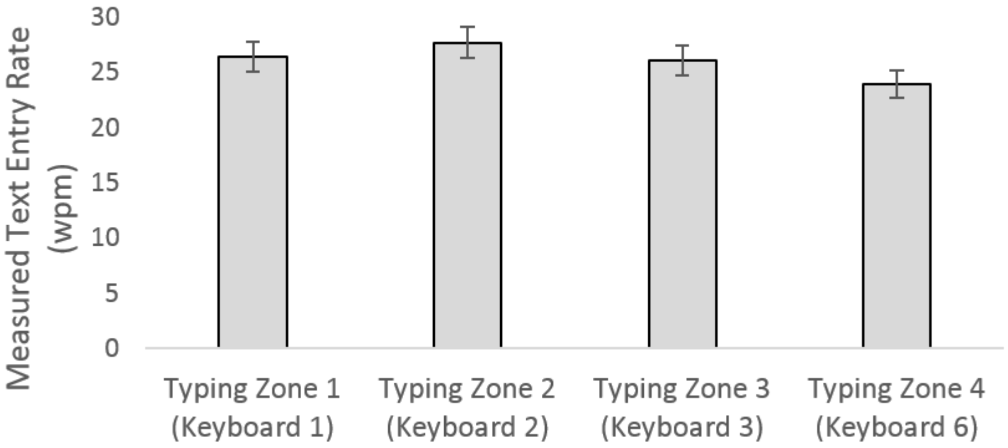


Figure 15. WPM for different typing zones under square keys (the rank is shown above the bar)

In Figure 16, the measured WPM for typing zone 2 under different *h/w* ratios are compared (keyboards 2, 4, and 5). Keyboard 5 has a statistically higher WPM than that of keyboard 4 ($p < .001$) supporting our observation in Figure 6 that the difference in WPM becomes smaller as *h/w* increases.

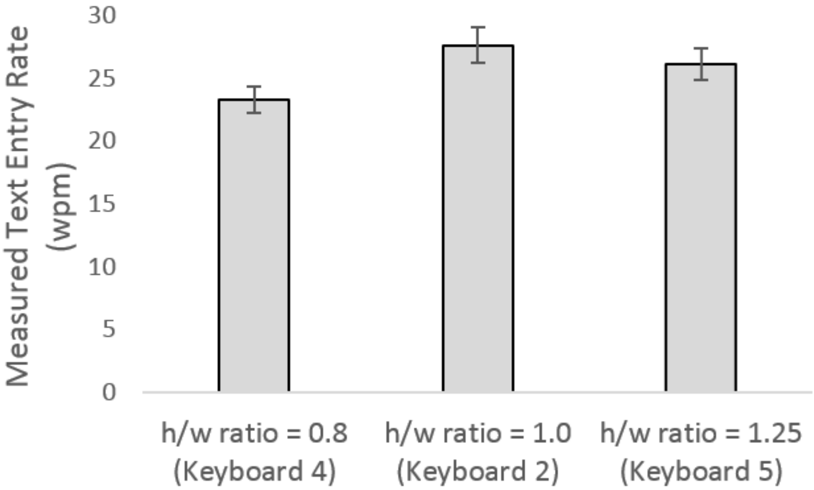


Figure 16. WPM for different h/w ratios under typing zone 2 (the rank is shown above the bar).

CHAPTER 5 DISCUSSION AND CONCLUSIONS

6.1 An Overview of the Contributions

A mixed integer-programming model is developed to optimize typing zones in two-thumb typing with the objective of minimizing the expected value of time to type a character. The model accounts for the occurrence and correction of typing errors as well as the alternate-hand advantage phenomenon and uses Fitts' law to estimate thumb movement times. Through computer simulation of the mixed integer-programming model for each of the different keyboard designs, the optimal split QWERTY keyboard in terms of key dimensions and typing zone is identified. The results show that a keyboard with square keys maximizes text entry rate. The results also suggest that the optimal typing zone depends on the keyboard dimensions and the user's speed-accuracy profile as well as the alternate-hand advantage.

The results of our experiments with eighteen users were used to empirically validate the analytical findings. On average, the text entry rate was 5% and 16% lower on the keyboards with a key height to width ratio of 1.25 and 0.8 than the proposed optimal keyboard with square keys, respectively. Therefore, the findings of our study provides helpful insights and general guidelines on the optimal design of split QWERTY keyboards on large touchscreen devices and thus can have significant implications for users, manufacturers, and researchers.

While the general findings from the mathematical model are consistent with the empirical results, theoretical WPM values predicted by the model are higher than the measured WPM values from the validation study. It is important to note that the primary goal of the mathematical model is to optimize typing zones in two-thumb typing by comparing the relative performance of different alternatives rather than trying to predict the actual text entry rates. While this difference can be due to the participants' lack of prior experience with split keyboards which can lead to slower movement times than the predicted values, improving finger movement time prediction models for touch-typing would be an important area for future research.

As noted in the Introduction, optimal two-thumb typing zones have not been considered in the literature with the exception of Negahban et al. (2013). It is important to note that the results of this work are more generalizable since we: (a) allow for the occurrence and correction of typing errors; (b) take into account different key dimensions; and (c), validate our results through an experimental study. The result of our paper shows that a slope coefficient of 0.127 would result in a theoretical optimal speed of 42.5 WPM, which is higher than the 36 and 37 WPM for the two layouts of Negahban et al. (2013). Although allowing for the occurrence and correction of typing errors decreases WPM, but the reason our WPM is higher is two-fold: (1) the value of the slope coefficient in our paper is smaller than the one of Negahban et al. (2013) (ours is a touch-typing parameter and theirs is a stylus-tapping parameter), which results in higher WPM; (2) the result of Negahban et al. (2013) is based on two fixed key dimensions only, whereas the result of our paper is optimized over different key dimensions as well, which results in higher WPM.

6.2 Limitations and Future Work

The proposed mathematical model involves several assumptions. Perhaps the most important limitation is that numbers, special characters, and punctuation marks are ignored. Therefore, an important extension involves expanding the current character transition frequency data to include these characters. We also assumed equal movement speeds for both thumbs regardless of movement direction. Moreover, several existing split keyboards in the market have a space key on each side and users have different strategies in terms of using one or both of the space keys. We consider only one space key where the mathematical model decides to which thumb (i.e., which side of the screen) it should be assigned. However, it is important to note that the proposed model is flexible in that the above considerations can be easily incorporated without making substantial changes to the model's general structure. Therefore, potential future extensions include relaxing some of the above assumptions as well as exploring other languages and keyboard layouts (e.g., quasi-QWERTY, Dvorak, FITALY, OPTI, Metropolis, and ATOMIK). While movement time may have a direct impact on health risks associated with excessive use of touchscreen keyboards, further research is needed to incorporate ergonomic considerations into the analysis.

For our validation study, we used a convenience sample of graduate students. Our hypothesis was that the mean age and its standard deviation would be higher than a convenience sample of undergraduate students and in the process make it more representative of the population of touchscreen tablet users. The participants in our study had a mean age of 29 years and a standard deviation of 3 years (i.e. the distribution covers users who are ~23-35 years of age). In future studies, it may be worth considering examining other age groups and/or users who come from different technical backgrounds (we assume that graduate students may be more technologically geared when compared to a general population). That being said, it is difficult to quantify whether our convenience sample affects the generalizability of the results since we could not find any published statistics on the demographics of touchscreen tablet users.

The authors believe that the use of touchscreen devices will continue to grow as their functionalities expand. We hope that this work and its extensions along the above lines will help improve the design of and user interaction with these popular devices in the future. To encourage future research in this area, we make our code and data freely available (see the Supplementary Materials for obtaining the links).

SUPPLEMENTARY MATERIALS

The optimization model was created using the IBM ILOG CPLEX Optimization Studio and Oracle's Java™ programming language and can be accessed through the following Github Repository: https://github.com/mza0052/Split-Keyboard. The data from the validation study is stored at Mendeley Data: https://goo.gl/evHdkW (doi:10.17632/7f3ky5ws8r.1).

REFERENCES

Alamdar Yazdi, M. A., Negahban, A., Cavuoto, L., & Megahed, F. M. (2018). Optimization of Split Keyboard Design for Touchscreen Devices. *International Journal of Human–Computer Interaction*, 1-10.

Azenkot, S., & Zhai, S. (2012, September). Touch behavior with different postures on soft smartphone keyboards. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services* (pp. 251-260). ACM.

Bi, X., Smith, B. A., & Zhai, S. (2010, April). Quasi-qwerty soft keyboard optimization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 283-286). ACM.

Bi, X., Smith, B. A., & Zhai, S. (2012). Multilingual touchscreen keyboard design and optimization. *Human–Computer Interaction, 27*(4), 352-382.

Colle, H. A., & Hiszem, K. J. (2004). Standing at a kiosk: Effects of key size and spacing on touch screen numeric keypad performance and user preference. *Ergonomics*, 47(13), 1406-1423.

Francis, G., & Oxtoby, C. (2006). Building and testing optimized keyboards for specific text entry. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 48*(2), 279-287.

Gao, Q., & Sun, Q. (2015). Examining the usability of touch screen gestures for older and younger adults. *Human Factors: The Journal of the Human Factors and Ergonomics*

Society, 57(5), 835-863.

Hsiao, H. C., Wu, F. G., Hsi, R., Ho, C. I., Shi, W. Z., & Chen, C. H. (2009). The evaluation of operating posture in typing the QWERTY keyboard on PDA. In *Ergonomics and Health Aspects of Work with Computers* (pp. 241-249). Springer Berlin Heidelberg.

Li, F. C. Y., Guy, R. T., Yatani, K., & Truong, K. N. (2011, October). The 1line keyboard: a QWERTY layout in a single line. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (pp. 461-470). ACM.

MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction, 7*(1), 91-139.

MacKenzie, I. S., & Soukoreff, R. W. (2002). A model of two-thumb text entry. *SPACE*, 67, 18-43.

MacKenzie, I. S., & Soukoreff, R. W. (2003, April). Phrase sets for evaluating text entry techniques. In *CHI'03 extended abstracts on Human factors in computing systems* (pp. 754-755). ACM.

MacKenzie, I. S., Zhang, S. X., & Soukoreff, R. W. (1999). Text entry using soft keyboards. *Behaviour & information technology*, 18(4), 235-244.

Madison, H., Pereira, A., Korshøj, M., Taylor, L., Barr, A., & Rempel, D. (2015). Mind the Gap The Effect of Keyboard Key Gap and Pitch on Typing Speed, Accuracy, and Usability, Part 3. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 57*(7), 1188-1194.

Marklin, R. W., Simoneau, G. G., & Monroe, J. F. (1999). Wrist and forearm posture from typing on split and vertically inclined computer keyboards. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 41*(4), 559-569.

Negahban, A., Uludag, I., & Yang, Y. (2013, January). An Integer Programming Approach to Optimize Typing Methods on Smartphones. In *Proceedings of the 2013 IIE Annual Conference* (p. 2657). Institute of Industrial Engineers (IIE), Norcross, GA.

Oulasvirta, A., Reichel, A., Li, W., Zhang, Y., Bachynskyi, M., Vertanen, K., & Kristensson, P. O. (2013, April). Improving two-thumb text entry on touchscreen devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2765-2774). ACM.

Pereira, A., Lee, D. L., Sadeeshkumar, H., Laroche, C., Odell, D., & Rempel, D. (2013). The Effect of Keyboard Key Spacing on Typing Speed, Error, Usability, and Biomechanics Part 1. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 55*(3), 557–566.

Pereira, A., Hsieh, C. M., Laroche, C., & Rempel, D. (2014). The Effect of Keyboard Key Spacing on Typing Speed, Error, Usability, and Biomechanics, Part 2 Vertical Spacing. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 56*(4), 752-759.

Quinn, P., & Zhai, S. (2016, May). A Cost-Benefit Study of Text Entry Suggestion Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (pp. 83-88). ACM.

Salthouse, T. A. (1986). Perceptual, cognitive, and motoric aspects of transcription typing. *Psychological bulletin, 99*(3), 303.

Sears, A., Revis, D., Swatski, J., Crittenden, R., & Shneiderman, B. (1993). Investigating touchscreen typing: the effect of keyboard size on typing speed. *Behaviour & Information Technology,* 12(1), 17-22.

Simoneau, G. G., & Marklin, R. W. (2001). Effect of computer keyboard slope and height on wrist extension angle. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *43*(2), 287-298.

Smith, A. L., & Chaparro, B. S. (2015). Smartphone text input method performance, usability, and preference with younger and older adults. Human Factors: *The Journal of the Human Factors and Ergonomics Society*, *57*(6), 1015-1028.

Soukoreff, R. W., & MacKenzie, I. S. (1995). Theoretical upper and lower bounds on typing speed using a stylus and a soft keyboard. *Behaviour & Information Technology, 14*(6), 370-379.

Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., & Shen, C. (2007, October). Lucid touch: a see-through mobile device. In *Proceedings of the 20th annual ACM symposium on User interface software and technology* (pp. 269-278). ACM.

Zhai, S. (2004). Characterizing computer input with Fitts' law parameters—the information and non-information aspects of pointing. *International Journal of Human-Computer Studies*, 61(6), 791-809.

Zhai, S., Hunter, M., & Smith, B. A. (2000, November). The metropolis keyboard-an exploration of quantitative techniques for virtual keyboard design. In *Proceedings of the 13th annual ACM symposium on User interface software and technology* (pp. 119-128). ACM.

Zhai, S., Hunter, M., & Smith, B. A. (2002). Performance optimization of virtual keyboards. *Human–Computer Interaction, 17*(2-3), 229-269.