Quasi-Qwerty Soft Keyboard Optimization

Xiaojun Bi
Department of Computer Science
University of Toronto, Toronto, Canada
xiaojun@dgp.toronto.edu

Barton A. Smith **Shumin Zhai**
IBM Research - Almaden
650 Harry Road, San Jose, CA, USA
{barton.smith, zhai}@acm.org

ABSTRACT

It has been well understood that optimized soft keyboard layouts improve motor movement efficiency over the standard Qwerty layouts, but have the drawback of long initial visual search time for novice users. To ease the initial searching time on optimized soft keyboards, we explored "Quasi-Qwerty optimization" so that the resulting layouts are close to Qwerty. Our results show that a middle ground between the optimized but new, and the familiar but inefficient (Qwerty) does exist. We show that by allowing letters to move at most one step (key) away from their original positions on Qwerty in an optimization process, one can achieve about half of what free optimization could gain in movement efficiency. An experiment shows that due to users' familiarity with Qwerty, a layout with quasi Qwerty optimization could significantly reduce novice users' visual search time to between those of Qwerty and a freely optimized layout. The results in this work provide designers with a new quantitative understanding of the soft keyboard design space.

Author Keywords

Soft keyboard, Qwerty, optimization, touch screens.

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms

Human Factors

INTRODUCTION AND RELATED WORK

To optimize or not to optimize has been a central question in soft keyboard layout design. In fact it has also been a central research question in physical keyboard design [5] although the current work is only focused on soft keyboards. Soft keyboards, also known as virtual, touch, graphical, on-screen or stylus keyboards, exist on all touch-screen devices. With the new generation of touch screen based mobile devices such as the iPhone coming to the market, soft keyboard design has an ever increasing relevance to users' everyday information interaction experience.

The starting point of a soft keyboard design is to mimic the de facto physical keyboard standard layout - Qwerty. Unfortunately Qwerty performs poorly as a soft keyboard since common consecutive letter pairs (digraphs) appear on the opposite sides of a keyboard [4]. When used with a single stylus or a single finger, back and forth lateral movement is more frequent and over greater distance than necessary. It is possible, but still not common, to tap a soft keyboard with two thumbs on a relatively large touch screen device.

Since at least the 1980's, researchers in accessibility and HCI have realized the shortcoming of Qwerty as a soft keyboard layout and designed a variety of optimized soft keyboard layouts with increasingly more sophisticated and more rigorous optimization methods. See [3, 4, 6] for just a few examples. These optimized layouts can lead to much higher movement efficiency over Qwerty, provided the user has learned the novel layouts.

However, learning is not what computer users are inclined to do in general even if its long term benefit clearly outweighs its short term investment. Researchers have also long realized this, and attempted to accommodate the initial learning in the keyboard design. For example to ease the initial learning process ATOMIK [6] is "alphabetically tuned" so that the optimized layout has an alphabetical tendency from one corner of the keyboard to another. But the success of such alphabetical tuning is rather limited, with about 10% speed increase during the very initial input phase. In fact even strictly alphabetical layout is not necessarily easy to learn either as a physical [5] or as soft [7] keyboard layout.

So, the chasm between optimizing for long term efficiency and designing for initial ease of visual search, which means Qwerty, is rather wide. The choice has been either there or here, either optimized or Qwerty.

A design space not well researched to date is whether a compromise can be found between the two goals: optimizing for movement efficiency while respecting existing computer users' familiarity with Qwerty. Specifically, we investigate the effects of rearranging key positions to minimize movement distance or time but within the following Qwerty constraints: the keyboard layout maintains Qwerty's 3 (row) by 10 (column) aspect ratio and each character can only be placed exactly at its corresponding position on the Qwerty or a position immediately adjacent to it. For example, the key "d" can only be placed at where "d" or "w", "e", "r", "f", "c", "x", "z", and "s" is on Qwerty. The possible benefits of such

CHI 2010, April 10–15, 2010, Atlanta, Georgia, USA.
Copyright 2010 ACM 978-1-60558-929-9/10/04....$10.00.

a middle ground design are two: higher movement efficiency than Qwerty and better initial visual search performance than free optimization. The potential downsides are also two: not much movement efficiency increases due to the very limited amount (one step) of movement from Qwerty and poor initial visual search performance because the letters are moved from Qwerty. Without research, we do not know the degree of either aspect.

At least two lines of previous work are related to the current investigation. The first is the work of iQwerty by Zhai and Kristensson [7]. As shown in Figure 1, iQwerty breaks each row of keys on Qwerty into two interlaced rows and still enabled users to leverage their familiarity with the Qwerty keyboard to visually locate letters even though they might be slightly away from their original positions. Second, users' familiarity of Qwerty has also motivated researchers to change the letter assignment on the 3 by 4 phone keypads to be "Qwerty-like"[1]. For example the three top row buttons on the 3 by 4 keypad in [1] were assigned with the letters "ewq", "tfy", and "op". These letters were close to but deviating from Qwerty in order to give better performance for multi-tap. Hwang and Lee's study shows that the similarity of such a layout to Qwerty could still enable faster learning and higher multi-tap text entry performance than a ABC layout "[1]. In fact a company has been built to commercialize such a design (mobience.com).

b	d	k	g		
c	a	n	i	m	q
f	l	e	s	y	x
j	h	t	o	p	v
		r	u	w	z

q	e	t	u	o
w	r	y	i	p
a	d	g	j	l
s	f	h	k	
z	c	b	m	
x	v	n		

Figure 1. Versions of ATOMIK (left) and iQwerty (right) layouts

However, neither of these lines of work has systematically and mathematically explored the optimization of soft keyboard with Qwerty constraint. Exploring this middle ground between free optimization on one end and complete status quo on the other has both theoretical and practical significance. Theoretically, it will help to understand performance optimality of text entry, a core interaction task, and its tradeoff with learning. Practically, it can lay a scientific foundation for industry designers to understand the design space and make an informed choice when placing a design on the spectrum. Without scientific research, such a decision would not have a solid foundation.

OPTIMIZATION WITH QWERTY CONSTRAINT

Soft keyboard optimization through the Metroplis energy minimization algorithm has been well documented in previous research reported in the literature [6]. For brevity we do not repeat any details of the method in this research note. Essentially, the algorithm picks two random keys from a current keyboard configuration and swaps their positions to reach a new configuration. The "Fitts-digram energy" (also called movement efficiency hereafter), defined as the sum of Fitts' law movement time between all pairs of letters (digrams) weighted by digram probabilities calculated from a language corpus, is then estimated. Whether the new configuration is kept as the starting position for the next iteration depends on the Metropolis function which guarantees that the search process can climb out of local minimums and approaches the lowest energy state after a sufficient number of iterations. It is important to note that the Fitts' law parameters (*a* and *b*, as in $MT = a + b\ ID$) are revised from what was reported in [6] to $a = 0.083$ sec and $b = 0.127$ sec based on measurements reported in more recent empirical work in the context of stylus keyboarding [8]. For the same amount of movement distance reduction in a new layout, these parameters will predict less time reduction than those reported in studies such as [6].

The statistical Metropolis optimization process can produce different layouts with very similar levels of mean movement time (or mean distance). We contrast and compare two specific layout instances produced from our optimization process, one with and one without Qwerty constraints. As shown in Figure 2, we denote the Qwerty layout as *Qwerty*, the layout optimized without Qwerty constraint as *Freely Optimized*, and the layout optimized with Qwerty constraint as *Quasi-Qwerty*. We also compare their movement efficiency for entering English with ATOMIK and iQwerty, two previously known layouts (Figure 1). The ATOMIK layout is adapted from ShapeWriter's iPhone app store release (cf www.shapewrtier.com) which has a number of changes from early versions of ATOMIK as reported in [6].

	k	j	z	x	
	f	c	h	t	w
q	u	o	i	s	p
y	m	a	n	e	r
	b	l	g	d	v

Freely Optimized

q	w	d	r	t	u	y	l	k	p
z	a	s	e	h	n	i	o	m	
	x	f	v	c	g	b	j		

Quasi-Qwerty

q	w	e	r	t	y	u	i	o	p
a	s	d	f	g	h	j	k	l	
	z	x	c	v	b	n	m		

Qwerty

Figure 2. Layouts of Freely Optimized, Quasi-Qwerty and Qwerty layouts

As Fitts-digram predictions show in Figure 3, *Freely Optimized* is much more efficient than *Qwerty*. It is also slightly more efficient than ATOMIK since it did not consider the alphabetical tuning that ATOMIK did. *Quasi-Qwerty* is more efficient than *Qwerty* but less than *Freely Optimized*. In fact it is about exactly half way between the two (*Freely Optimized* and *Qwerty*). It is also more efficient than iQwerty, which was designed more as a gesture keyboard for shape writing than a typing keyboard. Percentage wise, *Quasi-Qwerty* is 12% and 7% faster than *Qwerty* and iQwerty respectively, but 10% and 8% slower than *Freely Optimized* and ATOMIK respectively. If we evaluate the layouts by distance rather time, the mean movement distance to enter a letter on *Qwerty*, *Freely Optimized* and *Quasi-Qwerty* are 3.31, 1.88, and 2.54 keys respectively.

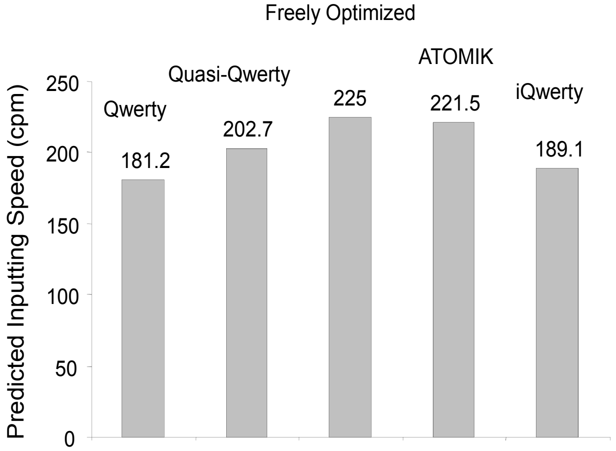

Figure 3 Movement efficiencies of five layouts

It is not surprising that *Quasi-Qwerty* is not as efficient as *Freely Optimized*. In fact it is somewhat surprising that by allowing only one step (one key) movement from Qwerty, a 12% reduction in time, and 23.3% reduction in distance could be achieved. Because of the intensity and frequency of text entry in daily information tasks, 10~20% reduction might have rather large practical significance.

INITIAL ENTRY TIME

Now that we know quantitatively the relative movement efficiency gain (over *Qwerty*) and loss (over *Freely Optimized*) of "near Qwerty optimization". We turn to the other side of soft keyboard usability: novice users' ability to visually locate the keys quickly. For that purpose we conducted an empirical experiment to measure the initial entry time, which is dominated by visually locating keys, of *Freely Optimized*, *Quasi-Qwerty*, and *Qwerty*.

Method

The experiment was conducted on a Lenovo X60 tablet PC equipped with a 12.1 inch 1024 X 768 pixels TFT screen with stylus input. The X60 screen is 245 mm wide and 184 mm high. Soft keyboards displayed on the screen were all set at the same key size. The center-to-center distance between adjacent square keys was 40 pixels.

Participants were asked to tap a list of 19 English words using the tablet digital pen on the soft keyboard displayed on the screen. The 19 words were tested in random order. If mistakes were made, the user had to click a button to clear it, and tap the word again. The list of the 19 words, first proposed by Zhai and Kristensson in [7], were "*the and you that is in of know not they get have were are bit quick fox jumps lazy*". These words cover all letters from A to Z, approximate letter frequencies in common English, and also approximate the English letter transition frequencies.

Twelve volunteers (4 female, 8 male), 26 ~ 55 years old, participated in the experiment. All were daily users of computers with Qwerty keyboards and all were right-handed.

Design

A within-subjects design with repeated measures was employed. The independent variable was soft keyboard layout (*Freely Optimized*, *Quasi-Qwerty*, and *Qwerty*). Participants tested all three layouts, whose order of appearance was balanced using a Latin Square. Prior to performing the tasks, participants tapped one English word that was not included in the experimental word list to familiarize themselves with the experimental procedure. During the study, they were instructed to perform the task as quickly and accurately as possible. Breaks were enforced between changes of soft keyboard layouts. The experiment lasted approximately 20 minutes for each participant.

The dependent variable was the time elapsed from the moment a word appeared on the screen until the last letter of this word was tapped. Since this was the very initial tapping experience, it primarily reflects the *initial visual search time* on *Freely Optimized* and *Quasi-Qwerty*.

Results

The mean *initial entry time* per word was 2110 ms (SD 492) on *Qwerty*, 3234 ms (SD 732) on *Quasi-Qwerty*, and 4705 ms (SD 1723) on *Freely Optimized* respectively (see Figure 4). Repeated measure variance analysis showed that layout ($F_{2,22}$=33.5, p<.0001) had a significant impact on the dependent variable.

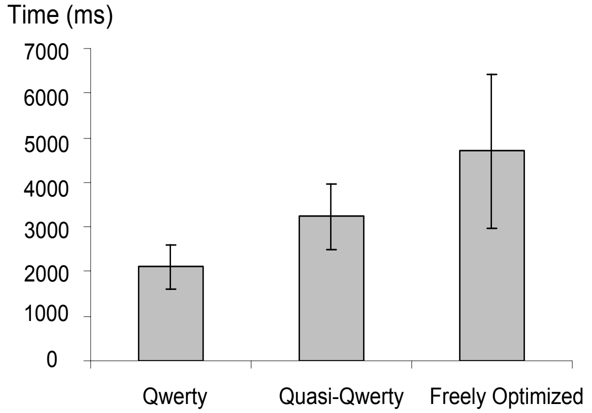

Figure 4. Mean and SD of initial entry time per word

Pairwise means comparison (Fishers LSD test) also showed a significant difference between every pair of layouts (p<0.0001). The mean (SD) error rates, the percentages of erroneous trials, were 1.4% (2.5%), 2.9% (2.8%), 2.3% (3.8%) for *Freely Optimized*, *Qwerty* and *Quasi-Qwerty*, respectively. Only correct trials were included in initial entry time analysis.

The result shows that the initial entry time of *Quasi-Qwerty* was about half way between *Freely Optimized* and *Qwerty*. The near Qwerty locations of the keys on *Quasi-Qwerty* certainly helped novice users to find them more quickly than on *Freely Optimized*, although the users were not nearly as fast as they were with the unaltered Qwerty.

Previous study [7] reported that ATOMIK and iQwerty layouts took 76.6% and 12.7% longer time than *Qwerty* in initial visual search for the same 19 words. iQwerty is also a quasi Qwerty design, but the change there is not optimized for movement efficiency. Rather it was a more systematic change hence possibly more predictable by the user.

DISCUSSION AND CONCLUSIONS

Although increasingly important due to the rapidly growing number of touch screen devices including mobile phones, tablets, GPS navigation devices and large surface computers, what layout of soft keyboards to use is still an open question. On one hand optimized layouts can save the users tremendous amount of time and effort in entering text on these devices in the long run. On the other hand, current computer users' familiarity with Qwerty means they would have to invest a few hours of accumulated use to surpass their performance on Qwerty as a software keyboard (See [4, 8] for empirical learning curves of new layouts). The industry to date has been stuck with two choices, either the familiar status quo of Qwerty, or freely optimized but completely novel layouts. The current work shows that there is a quasi-Qwerty optimization middle ground between these two ends. The quantitative findings in the present work provide a foundation for soft keyboard design. Where to place a commercial design on the Qwerty - quasi Qwerty - free optimization spectrum can be an informed choice depending on the product design criteria. If a few hours of learning is not an issue and what matters is efficiency after some training, then fully optimized layouts are the rational choice. If the user's immediate experience is all that matters, then Qwerty should definitely be the approach. If one wants to accommodate some of both ends, then there is indeed such a middle ground. Note that what we have experimented is just one point in the middle spectrum. Conceivably we could also have a different degree of similarity to Qwerty. To one side for greater familiarity we could place more stringent Qwerty constraint by allowing a letter to be move only in some directions (e.g. laterally only). To the other side we could also allow some letters to move more than one key for greater efficiency increase. The benefits and drawbacks of such designs can be studied with the same methodology developed in the current work.

REFERENCES

1. Hwang, S. and Lee, G., Qwerty-like 3x4 keypad layouts for mobile phone. *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, ACM, New York, NY, 1479-1482.
2. Lewis, J. R. Typing-key layouts for single-finger or stylus input: initial user preference and performance, International Business Machines Corporation, Boca Raton, FL (1992).
3. Lewis, J. R., Kennedy, P. J. and LaLomia, M. J. Improved typing-key layouts for single-finger or stylus input, IBM Technical Report TR 54.692 (1992).
4. MacKenzie, I. S. and Zhang, S. X., The design and evaluation of a high-performance soft keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*,(CHI'99). ACM, New York, NY, 25-31.
5. Norman, D. A. and Fisher, D. Why alphabetic keyboards are not easy to use: Keyboard layout doesn't much matter. *Human Factors*, 24, 5 (1982). 509-519
6. Zhai, S., Hunter, M. and Smith, B. A., 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* (UIST'00), ACM, New York, NY, 119-218.
7. Zhai, S. and Kristensson, P. O., Interlaced QWERTY - accommodating ease of visual search and input flexibility in shape writing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (CHI'08), ACM, New York, NY, 593-596.
8. Zhai, S., Sue, A. and Accot, J., Movement model, hits distribution and learning in virtual Keyboarding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (CHI'02) ACM, New York, NY, 17-24.