Computational Layout Design for Keyboards with Multi-Letter Keys

Ryan Qin
Ward Melville High School
East Setauket, NY, USA
Stony Brook University
Stony Brook, NY, USA
ryanqin15@gmail.com

Yu-Jung Ko
Computer Science Department
Stony Brook University
Stony Brook, NY, USA
yujko@cs.stonybrook.edu

Suwen Zhu
Computer Science Department
Stony Brook University
Stony Brook, NY, USA
suwzhu@cs.stonybrook.edu

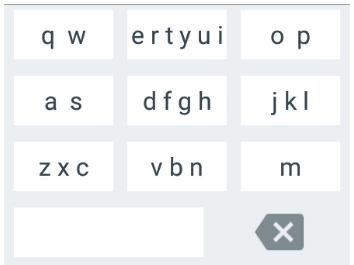
Xiaojun Bi
Computer Science Department
Stony Brook University
Stony Brook, NY, USA
xiaojun@cs.stonybrook.edu

Yu-Hao Lin
Computer Science Department
Stony Brook University
Stony Brook, NY, USA
yuhalin@cs.stonybrook.edu

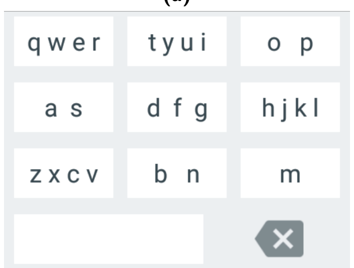
Abstract
Keyboards with multi-letter keys (i.e., a key corresponds to multiple letters) have been commonly used on small touchscreen devices to mitigate the problem of tapping tiny keys with imprecise finger touch (e.g., T9 keyboard). We have proposed a computational approach to designing optimal multi-letter key layouts by considering three key factors: clarity, speed, and learnability. In particular, we have devised a clarity metric to model the word collisions (i.e., words with identical tapping sequences), used the Fitts-Digraph model to predict speed, and introduced a Qwerty-bounded constraint to ensure high learnability. Founded upon rigorous mathematical optimization, our investigation led to Qwerty-bounded T9-like (i.e., 3×3) layouts optimized for both clarity and speed. A preliminary user study showed promising performance of such keyboards.

ACM Classification Keywords
H.5.2. [Information Interfaces and Presentation]: User Interfaces-Input devices and strategies.

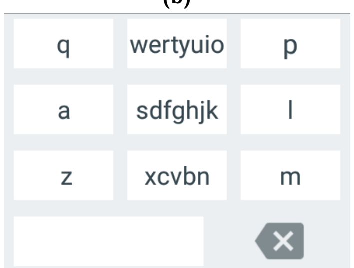
Author Keywords
Text entry; touchscreen.



(a)



(b)



(c)

Figure 1: (a): the optimized layout that maximizes the average of the clarity and speed scores. (b): the layout with maximum clarity score. (c): the layout with maximum speed score.

Introduction

Typing on keyboards supported by small embedded devices such as smartwatches is often extremely cumbersome. As the finger is inherently inaccurate, the combination of an imprecise input device and tiny congested keys makes typing incredibly error-prone. One of the most popular approaches to combat this input problem is via a multi-letter key design, in which individual letters are amalgamated to enlarge the key size.

We advocate a novel computational approach for designing multi-letter key layouts by considering three important factors in layout design: clarity (i.e., reducing the number of words with identical tapping sequences), speed, and learnability. In particular, we have devised a clarity metric to model the word collisions (i.e., words with identical tapping sequences), used the Fitts-Digraph model [2, 14] to predict speed, and introduced a Qwerty-bounded constraint to ensure high learnability. Based on the proposed models, we applied a rigorous mathematical optimization with a Qwerty-bounded constraint to search for optimal 3×3 multi-letter layouts. To understand to what degree the optimized layout would improve typing performance in realistic text entry tasks, we conducted a pilot study to evaluate the performance of the optimized layout alongside two de facto standard layouts : Qwerty and T9.

Related Work

Various keyboard optimization approaches have been proposed, beginning with improving input speed exclusively [2] to eventually considering multiple factors such as speed, accuracy, and learnability [1, 3, 4] with both single-letter and multi-letter key layout design. Methods including the Metropolis algorithm [13], Pareto multi-objective optimization [12], and integer programming [9] have been proposed based on single-letter key layout design. Alphabetically constrained keypads [5] and the Qwerty-like 9-key layout [7], a multi-letter key layout optimized with a bias for layout adaptability, have also been introduced. Moving forward, we advance the multi-letter layout optimization to performing Pareto optimization on three critical objectives – speed, accuracy, and learnability.

Past research has also shown that the modern statistical decoding technique worked reasonably well on small keyboards. Gordon et al.'s work [6] revealed that human motor control adaptability, coupled with modern statistical decoding and error correction technologies developed for smartphones, can enable a surprisingly effective typing performance for both gesture typing and tap typing on a regular Qwerty keyboard on a watch-sized screen. Inspired by Gordon et al.'s research, we coupled multi-letter key layout design with the modern statistical decoding technology and compared the optimized multi-letter key layout with a regular Qwerty keyboard.

Optimizing Multi-Letter Key Layouts

A number of factors must be carefully considered and balanced in the keyboard design task. For a novel layout to flourish, we believe the following factors are key: clarity, speed and learnability.

Clarity defines a multi-letter key layout's capability of minimizing the potential word collisions (i.e., words sharing identical tap sequences because of merged keys). We define clarity score to describe how likely layout L can resolve word collisions:

$$C(L) = \sum_{j=1}^{M} f(W_j) clarity(W_j), \tag{1}$$

where M represents the number of words comprising the corpus, $f(W_j)$ is the frequency of a given word W_j, and $clarity(W_j) = \frac{f(W_j)}{\sum_{i=1}^{N} f_{W_i}}$ is a value between 0 and 1 for W_j among total number of words N. The corpus for optimization was taken from American National Corpus (ANC) [8].

The typing speed metric estimates how fast expert users will be able to tap type on a keyboard layout. We used the widely known Fitts-Digraph model [2, 14] for speed prediction, which shows that the average time (t) for inputting a letter is:

$$t = \sum_{i=1}^{26}\sum_{j=1}^{26} P_{ij}T_{ij}, \quad (2)$$

where P_{ij} is the frequency of the ordered character pair i, j from 26 Roman characters, and T_{ij} is the movement time for the input finger travelling from key i to key j, which is typically predicted by the Fitts' law:

$$T_{ij} = a + b\log_2(\frac{D_{ij}}{W_{ij}} + 1), \quad (3)$$

where D_{ij} is the distance from the center of key i to the center of key j, and W_{ij} is the key width. Since each key tap action is essentially a 2-dimensional Fitts' law task, we used $min(W_{ij}, H_{ij})$ (i.e., the minimum of key width or height) as W_{ij} in Equation (3) [11]. Previous research [11] showed it yielded a fairly successful fit for 2D Fitts tasks. In the context of touchscreen typing, Fitts' law parameters were $a = 0.083s$ and $b = 0.127s$, estimated by Zhai et al. [14]. t has the unit of seconds. t can be converted to input speed (V) in characters per minute (CPM): $V = 60/t$.

Learnability is critical to the success of any new layout design: perhaps the biggest obstacle of any newly optimized keyboard is learning the layout. Consequently, despite numerous layouts having been proposed, very few are actually implemented extensively. To achieve superior performance over existing layouts, users likely have to spend a considerable amount of time practicing, and not every user is willing to make such an effort. For an optimal layout to maintain high learnability, we devise a strict Qwerty-bounded constraint: we preserve Qwerty's alphabetical arrangement to ensure that users can immediately use this keyboard fluently. Note that the Qwerty-bounded constraint only works for layouts with 3 rows.

Multi-Objective Optimization
With the two aforementioned objectives (clarity and speed) and the Qwerty-bounded constraint (learnability), designing a multi-letter key layout is essentially a multi-objective optimization problem: searching for a layout optimized for both clarity and speed, subject to the Qwerty-bounded constraint.

As commonly used in layout optimization research, we adopted the Pareto optimization technique [4, 3] to address this multi-objective optimization problem. Instead of generating a single optimized layout, Pareto optimization will lead to a Pareto front, in which each layout is Pareto optimal, meaning that none of its metric scores can be improved without compromising the other scores. The designer then later picks layouts from the Pareto front, after considering the relative weights between metrics or other factors.

Computationally Designing 3×3 Layouts
Our algorithmic overview consists of the following three major phases. First, we exhaustively iterate through all layout candidates subject to the Qwerty-bounded

constraint. Second, we utilize the Pareto optimization approach to attain the final optimal configuration. Third, to empirically evaluate the proposed computational approach, we applied it to design optimal 3×3 layouts for a watch-size multi-letter key layout based on Apple Watch screen specification (312 pixels (26.15 mm) by 390 pixels (32.69 mm)). A watch platform was selected as we devise this novel computational approach with the aim of improving text entry specifically on small devices.

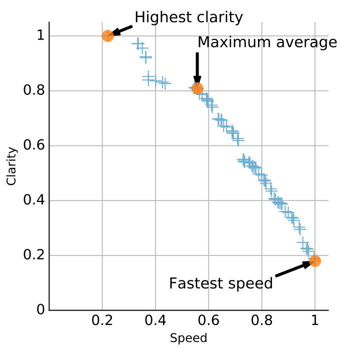


Figure 2: The Pareto front.

Figure 2 illustrates the complete Pareto front formed by 71 Pareto optimal layouts. As shown, the front approximately forms a curve spanning the top-left and bottom-right corner, indicating that clarity and speed are conflicting metrics: one metric increases at the expense of the other. Figures 1b and 1c display the layouts at two ends of the front: the one possessing the highest clarity and the one holding the fastest speed. In Pareto optimization, the final compromise keyboard proposal is taken to be the keyboard that achieves best on average. Thus, as we are particularly interested in the layouts with the most balanced typing clarity and speed, we closely examine the layouts near the center of the Pareto front. We selected the layout carrying the maximum average of normalized clarity and speed as the optimized layout subject to our specific Qwerty constraints. We referred to this configuration as our optimized layout (Figure 1a), which lies on the 55.4 degree line from the origin. The clarity scores and estimated input speeds are shown in Table 1.

	Optimized	Highest clarity	Fastest speed	T9	Qwerty
Clarity	0.8738	0.9412	0.6519	0.9234	1.0
CPM	309.70	284.27	343.14	278.18	169.74
WPM	61.94	56.85	68.63	55.64	33.95

Table 1: The clarity and speed (in CPM and WPM) of different 3×3 layouts.

Evaluation

We carried out a preliminary study with 4 users (1 female) aged from 25 to 34. The average text entry speed following Mackenzie [10] was 18.99 WPM ($SD = 4.03$) for the optimized keyboard, 14.76 WPM ($SD = 0.43$) for T9, and 18.19 WPM ($SD = 2.93$) for Qwerty. Additionally, the average word error rate was 2.05% ($SD = 1.54\%$) for the optimized keyboard, 2.30% ($SD = 0.64\%$) for T9, and 1.54% ($SD = 1.41\%$) for Qwerty. At the end of the study, participants were asked to give an overall subjective rating for each keyboard on a continuous scale of 1 (very dislike) to 5 (very like). The average rating was 4.5 for the optimized keyboard, 1.75 for T9, and 3.75 for Qwerty.

Overall, the small-scale study results showed the optimized layout was promising. Its input speed was greater than original T9 and Qwerty, and the subjective ratings were also in favor of it. We plan to carry out a more formal and large-scale user study to investigate its performance.

Conclusions and Future Work

We have proposed a computational approach for designing optimal multi-letter key layouts by taking into consideration clarity, speed and learnability. To evaluate its validity, we have applied it to computationally design 3×3 layouts. Our investigation led to an optimized layout which struck a balance between clarity and speed. Both the theoretical analysis and a preliminary user study showed such a layout has outperformed the original T9 layout and could be promising for text entry on small touchscreen devices (e.g., smart watches). We plan to carry out more formal studies to further investigate the pros and cons of the proposed methods as well as the generated optimal keyboard layouts.

References

[1] Bi, X., Smith, B. A., and Zhai, S. Quasi-qwerty soft keyboard optimization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, ACM (New York, NY, USA, 2010), 283–286.

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

[3] Bi, X., and Zhai, S. Ijqwerty: What difference does one key change make? gesture typing keyboard optimization bounded by one key position change from qwerty. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, ACM (New York, NY, USA, 2016), 49–58.

[4] Dunlop, M., and Levine, J. Multidimensional pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, ACM (New York, NY, USA, 2012), 2669–2678.

[5] Gong, J., and Tarasewich, P. Guidelines for handheld mobile device interface design. In *In Proceedings of the 2004 DSI Annual Meeting* (2004).

[6] Gordon, M., Ouyang, T., and Zhai, S. Watchwriter: Tap and gesture typing on a smartwatch miniature keyboard with statistical decoding. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, ACM (New York, NY, USA, 2016), 3817–3821.

[7] Hwang, S., and Lee, G. Qwerty-like 3x4 keypad layouts for mobile phone. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, ACM (New York, NY, USA, 2005), 1479–1482.

[8] Ide, N., and Macleod, C. The american national corpus: A standardized resource of american english. In *Proceedings of Corpus Linguistics*, Vol. 3 (Lancaster, UK, 2001), 1–7.

[9] Karrenbauer, A., and Oulasvirta, A. Improvements to keyboard optimization with integer programming. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, ACM (New York, NY, USA, 2014), 621–626.

[10] MacKenzie, I. S. A note on calculating text entry speed, 2015.

[11] MacKenzie, I. S., and Buxton, W. Extending fitts' law to two-dimensional tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, ACM (New York, NY, USA, 1992), 219–226.

[12] Smith, B. A., Bi, X., and Zhai, S. Optimizing touchscreen keyboards for gesture typing. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, ACM (New York, NY, USA, 2015), 3365–3374.

[13] 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, USA, 2000), 119–128.

[14] 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, USA, 2002), 17–24.