Multilingual Touchscreen Keyboard Design and Optimization

Xiaojun Bi

University of Toronto

Barton A. Smith

IBM Almaden Research Center

Shumin Zhai

Google

RUNNING HEAD: MULTILINGUAL TOUCHSCREEN KEYBOARDS

Corresponding Author's Contact Information:

Shumin Zhai

zhai@acm.org

408-927-1112

Brief Authors' Biographies: (follow the template below)

Xiaojun Bi is a computer science researcher with an interest in Human Computer Interaction; he is a PhD candidate in the Department of Computer Science at University of Toronto. **Barton A. Smith** is a scientist with an interest in helping people use information technology to do their work; he is a Research Staff Member in the Computer Science department at the IBM Almaden Research Center. **Shumin Zhai** is a research scientist with an interest in foundational issues in human-computer interaction and user interface innovation. He recently joined Google after 15 years of service as a Research Staff Member in the Computer Science department at the IBM Almaden Research Center.

ABSTRACT

A keyboard design, once adopted, tends to have a long lasting and world wide impact on daily user experience. There is a substantial body of research on touchscreen keyboard optimization. Most of it has focused on English only. Applying rigorous mathematical optimization methods and addressing diacritic character design issues, this paper expands this body of work to French, Spanish, German, and Chinese. More importantly and counter to the intuition that optimization by nature is necessarily specific to each language, this paper demonstrates that it is possible to find common layouts that are highly optimized across multiple languages. Applying a multilingual optimization method, we first obtained a touchscreen keyboard layout that is highly optimized for both English and French input. We then obtained a layout that is optimized for English, French, Spanish, German and Chinese *pinyin* simultaneously, reducing its stylus tapping travel distance to about half of QWERTY's for all of the five languages. In comparison to QWERTY's 3.31, 3.51, 3.7, 3.26, 3.85 keys of movement for English, French, Spanish, German and Chinese respectively, the optimized multilingual layout has an average travel distance of 1.88, 1.86, 1.91, 1.77 and 1.68 keys correspondingly. Applying Fitts' law with parameters validated by a word tapping experiment, we show that these multilingual keyboards also significantly reduce text input time for multiple languages over the standard QWERTY for experienced users. In comparison to layouts individually optimized for each language, which are also obtained in this paper, simultaneously optimizing for multiple languages caused only a minor performance decrease for each language. This surprising result could help to reduce the burden of multilingual users having to switch and learn new layouts for different languages. Additionally, we also present and analyze multiple ways of incorporating diacritic characters on multilingual keyboards. Taken together, the present work provides a quantitative foundation for the understanding and designing of multilingual touchscreen keyboards.

CONTENTS

1. INTRODUCTION

Few industrial designs have been as consequential and long lasting as the QWERTY keyboard. Amid many competing inventions of the writing machine in the 1850's and 1860's, the typewriter designed by Christopher L. Sholes, Carlos Glidden and Samuel W. Soule, considered the 52nd by some (Yamada, 1980), eventually took hold. The underlining mechanisms of the typewriter have completely changed from one generation to another many times since then but the QWERTY layout has persisted for almost one and half centuries. Even more remarkable is that the consequence of this design has only increased dramatically in the last few decades in the number of users and in geography. Typing used to be the work of professional typists or secretaries in European and American countries, but today it is a daily activity of billions of world-wide computer users. Almost every computer user is affected by the QWERTY keyboard each time he or she touches a computer. The fascinating and controversial history of the typewriter and the QWERTY keyboard is the subject of many books and articles (Yamada, 1980).

Particularly relevant to the motivation of the current work are two important lessons. One is the sheer amount of impact of a keyboard design on the experience of so many users and the other is the cultural and linguistic aspects of its design consequences.

Suppose a keyboard design saves a centimeter of finger travel per letter (hence about five centimeters per word) and if an active user types on average 100 words (of email, texting, blogging etc) per day, then 100 (words) X 5 (cm) X 365 (day) = 1.825 kilometers of travel will be saved per active user per year. Assuming that the total number of active computer users is one billion (the total number of internet users is estimated at nearly two billion today according to http://www.internetworldstats.com/stats.htm), then a total of 1.825 billion kilometers of movement would be saved per year! Of course this estimation exercise is only an illustration of the potential impact of basic user interface design magnified by the ubiquity of computers today. What really matters to text writing is the total user experience including the cognitive, perceptual, motor and affective elements. However, in text input research, the focus has been and continues to be on time efficiency due for two reasons. One is that it is much more difficult to understand and quantify user experience other than by time and accuracy. Indeed the controversies in the history of the typewriter often have to do with the fact that different design and evaluation criteria were used by different researchers. Some might have designed for speed and accuracy. Others might have focused on legibility, fatigue, and learning (Yamada, 1980). The second reason, often unstated, is that speed, accuracy and learning may ultimately be good indicators of higher order experience. For example, an important advantage of touch typing lies in the perceptual effort savings because one does not have to switch visual attention between the keyboard and the written text; but such an advantage may be reflected in a greater amount of text entered with fewer errors uncorrected per unit time. Fun or enjoyable experience is very important to product design; but an efficient method may well be ultimately more fun and more enjoyable than a less efficient one. Nevertheless it is important to keep in mind that physical efficiency should be treated as an indicator of interaction quality not the ultimate design goal. If better user experience can be found in a design that at odds with efficiency, the former should be given greater consideration.

Although often ignored, there should be important cultural and linguistic consideration in user interface design (del Galdo & Nielsen, 1996; Shneiderman, 2000). These considerations are even more obviously important in text writing interfaces because users in different parts of the world speak and write in different native languages. While ideally, user interface designs should be based on the unique cultural and linguistic characteristics in each region, we reason that it is even better to develop common designs that simultaneously optimize for multiple geographies and different user populations. This is because technologies tend to spread from one region to another while demanding compatibility. Take the QWERTY keyboard design as an example. One would expect very different designs be made in different countries since the QWERTY design was based on the English language (Yamada, 1980), but in fact only slight variations of the QWERTY were made in for example French and German speaking countries (http://en.wikipedia.org/wiki/International_QWERTY_keyboards) and no changes made in China. It is interesting to note that as one of the most knowledgeable and motivated researchers, Prof. Yamada, wrote his historical study paper of typewriters "from the position of planning Japanese parallels" before the personal computer revolution that made typing an activity of everyone and everyday. However today QWERTY is still the standard keyboard layout in Japan. When a new product such as the PC enters a country, it is usually not planned introduction en masse. Rather it starts from a small group of highly technical users and spreads over time. At any given time those who already are familiar with the existing design would be more influential but less motivated to change the design. Therefore it is never a good time to put a stop on the spread of an existing design and do a total design recall. Another important factor to consider is that the force of technology can also change local cultures in surprisingly powerful ways. When one of the authors of this paper grew up in China in the 1960's and 1970's, China had undergone more than a century of "westernization" (or modernization) willingly or unwillingly. This included the attempts to reform its beloved logographic writing system. Despite these attempts the Romanized version of the Chinese writing system, *pinyin*, was only considered a tool for teaching phonetics to school pupils. Most school pupils would learn it and then forget it after they mastered enough number of Chinese characters. The consensus was that *pinyin* would never be more useful than a phonetics teaching tool because Chinese character are often homophonic and because people in different parts of China speak with very different accents or in different dialects that are not mutually intelligible. When the PC was introduced to China in the early1980's Chinese input systems based on the logographic character strokes were invented and heavily promoted, but few in China used computers for communications untill at least the late 1990's. Surprisingly, however, today the overwhelming majority of hundreds of millions of computer and mobile phone users in China use *pinyin* to enter Chinese characters. People speaking with different accents or dialects simply memorize the correct *pinyin* spellings. In other words, in just over a decade, the PC, the internet, and the mobile phone technologies have propelled the use of *pinyin*, or the Romanization of Chinese writing, to a height that scholars and language reformers have tried but failed to achieve in more than a century.

While the layout of the physical keyboard on computers is probably a "done deal" due to the large existing world-wide user base, the present work instead focuses on multilingual keyboard optimization for touch screens. Touchscreen keyboards, also known as soft, virtual, touch, graphical, on-screen or stylus keyboards, are a common part of user interfaces of the increasingly popular touchscreen devices. Typically these touchscreen keyboards use the familiar QWERTY layout. However, QWERTY, or its variations for non-English languages such as QWERTZ and AZERTY, perform poorly as a stylus or single finger keyboard (Lewis, 1992; Lewis, Kennedy, & LaLomia, 1999; Lewis, LaLomia, & Kennedy, 1999; MacKenzie & Zhang, 1999) since common consecutive letter pairs (known as digraphs or bigrams) tend to appear on the opposite sides of the keyboard (Yamada, 1980). When used with a single stylus or a single finger, back and forth lateral movement is more frequent and more distant than necessary on QWERTY. To address this shortcoming, a substantial amount of research on optimized touchscreen keyboard layouts has been done (Getschow, Rosen, & Goodenough-Trepagnier, 1986; Lewis, 1992; Lewis, Potosnak, & Magyar, 1997; MacKenzie & Zhang 1999; Soukoreff & MacKenzie, 1995; Zhai, Hunter, & Smith, 2000, 2002, (Dell'Amico, Díaz, Iori, & Montanari, 2009) with an increasing degree of sophistication and mathematical rigor. These efforts to date have focused on one language, namely English, despite the calls for international and universal design in the HCI community (del Galdo

& Nielsen, 1996; Shneiderman, 2000). There is a need to expand touchscreen keyboard optimization research to other languages, which the present work does to French, German, Spanish and Chinese.

It is well established that optimized touchscreen keyboards have a significant time and motion advantage over QWERTY (MacKenzie & Zhang, 1999; Zhai, Sue, & Accot, 2002) once the user learns the layout to the extent that the input performance bottleneck lies primarily in motor movement instead of visual search. The learning requirement, although only a matter of hours on average (MacKenzie & Zhang, 1999; Zhai, Sue, et al., 2002) is the main adoption obstacle to optimized layouts in comparison to the familiar QWERTY.

One advantage of touchscreen keyboards is that their layout is completely graphical and can be changed easily without material cost. The cost of having multiple layouts, however, lies in user learning. If keyboard layouts are specifically optimized for each language, multilingual users have to multiply their learning effort. Even if the user learns multiple layouts, switching between two or more layouts in use may also be a negative user experience.

Learning and using a second language is often a necessity in today's globalized world, particularly for professionals and business people outside of the English speaking countries. It is therefore highly desirable to have new keyboard layouts common across multiple languages.

However, the idea of optimizing for multiple languages seems to contain a contradiction, because keyboard optimization by definition means taking advantage of the uneven digraph frequencies in a particular language and placing the common digraphs (e.g., t-h in English) close to each other for that language. If we simultaneously optimize for multiple languages, the results could be, one would expect, significantly sub-optimal for each one of them.

Two observations led us to conjecture that these suboptimal layouts may not be much less optimized than the layout specifically optimized for one language. First, letter and digraph distributions in different languages may have a great deal of commonality, as we will quantitatively show later. Second and more importantly, past research experience (Zhai, Hunter, & Smith, 2002) in using energy minimization techniques to optimize touchscreen keyboard layouts shows that there is a wide range of layouts with similar levels of efficiency for a given language. In other words, the lowest "energy" configuration is not necessarily at the bottom of a V shaped canyon, but rather somewhere on a fairly wide U shaped valley floor. This observation led to the ATOMIK layout which uses the additional optimization space to accommodate additional design considerations such as alphabetical ordering and complete connection of common words (Zhai, Hunter, et al., 2002). Of course our conjecture based on these two observations is only speculative. Without mathematical analysis and computational research, we would never know the degree of a multilingual keyboard's optimality in comparison to a unilingual keyboard.

Designing for multiple languages has been previously attempted outside of the research literature. For example the AgileText (http://agiletext.com) layout was evaluated against the top 10 to 100+ words in various western languages. In this paper we develop rigorous and systematic methods of optimizing touchscreen keyboard layouts for multiple languages. We first optimize for both English and French, and then expand to Spanish, German, and even Chinese *pinyin* which shares less commonality with European languages. Our research demonstrates that (1) it is possible to significantly improve input speed for each of the five languages simultaneously over QWERTY, and (2) simultaneously optimizing for five languages only causes minor performance degradation compared to optimizing for each separately. We conducted an empirical experiment to verify the accuracy of a Fitts-digraph model and its parameters employed in the optimization process, which reinforces the theoretical predictions of keyboard performance. We also investigated how diacritic characters should be treated in optimized multilingual touchscreen keyboards.

2. BACKGROUND AND RELATED WORK

2.1 Input Methods for Different Languages.

The number of languages currently used in the world is between 6000 to 8000 (Gordon & Grimes, 2005). These languages belong to more than 90 different language families, that is, groups of languages sharing similar origins. The top 2 language families are Indo-European and Sino-Tibetan in terms of numbers of native speakers. Languages in the Indo-European family are spoken by almost 2.6 billion native speakers, around 44% of the world's population. They include most of the common languages in western countries such as English, German, French, and Spanish. The Sino-Tibetan language family has the second largest number of speakers, around 1.3 billion speakers or 22% of the world's population. The major language in this family is Chinese, which is spoken by more than 1 billion people.

How to effectively input various languages via a keyboard is an important topic of research and design. In general, it is common to modify the QWERTY keyboard to languages sharing a similar alphabet with English. Most of the letters in for example German, French and Spanish are already located on the basic QWERTY keyboard. Handling different diacritics unique to each language can be achieved by either adding separate keys (e.g., é and ç on an AZERTY) or using modifier keys (also known as Dead-Keys). Discussion of these two methods is detailed in Section 5 of this paper. Inputting languages which are fundamentally different from English, such as Chinese, raises greater design challenges. As alluded to in the introduction, there are presently two broad categories of keyboard-based methods for entering Chinese characters. One is by typing *pinyin*, either on a QWERTY layout or an alphanumeric keypad on traditional mobile phones. Chinese characters corresponding to the *pinyin* stream entered are then displayed and selected by the user. The homophonic character selection process may take as much time as the pinyin typing process (Wang, Zhai, & Su, 2001). The other keyboard-based method is to encode the shape of the logographic characters according to their strokes and structures, with keys on a keyboard. This method is often preferred by professional typists due to its faster input speed because fewer key strokes are needed per Chinese character. However, the downside is that it imposes memory burdens on novice users to learn the key mappings.

Although a sizeable amount of research has been conducted about inputting non-English languages on a keyboard, most of it focuses on inputting one specific language only. See (MacKenzie & Tanaka-Ishii, 2007) for a broader review. We take a different direction in this paper: exploring how to accommodate multiple languages simultaneously.

2.2 QWERTY layout

QWERTY, designed by Christopher L. Sholes and colleagues in 1867 (Yamada, 1980), is the de facto standard for physical keyboards. Although designed for English input, it is also used to input other languages. In some countries, QWERTY is slightly modified to suit their languages. In French-speaking countries, the AZERTY layout, on which A and Q, Z and W are switched from their QWERTY positions, is commonly used. The QWERTZ keyboard, on which Y and Z are switched, is used in Germany and much of central Europe. Note that shown in Figure 1 the keys are aligned in horizontal rows and vertical column. In mechanical typewriters columns are not vertical but diagonal in order to give space to the levers holding each key in the same column. This is no clear reason for such a diagonal alignment in touchscreen keyboards.

Figure 1. QWERTY keyboard

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**		

Although the QWERTY layout (Figure 1) was designed to minimize typewriter mechanical jamming by arranging common digraphs on the opposite sides of the keyboards (Yamada, 1980), it also works well for two handed typing because it facilitates frequent alternation of the left and right hand. As a single movement point (finger or stylus) keyboard, it has been long understood as inefficient. Starting from at least as early as Getschow and colleagues' optimization work for increasing efficiency for the motor impaired (Getschow, Rosen, & Goodenough-Trepagnier, 1986; Lewis, Kennedy, et al., 1999), researchers and developers have tried to find various ways to design more efficient alternatives, first using simple algorithms (Getschow, et al., 1986; Lewis, Kennedy, et al., 1999) or heuristics (MacKenzie & Zhang, 1999), eventually to more rigorous mathematical optimization (Zhai, Hunter, & Smith, 2000; Zhai, Hunter, et al., 2002). However optimization of touchscreen keyboards to date has focused on single language designs only.

2.3 Optimization Objectives

Quantitative optimization is only possible with a well defined objective function. Interweaved with thinking, composition, and visual search or visual verification, text entry is a complex task involving cognitive, perceptual and motor components. However after a sufficient amount of learning, typing performance is limited primarily by hand movement on the keyboard. Touchscreen keyboard optimization work has therefore focused on movement time reduction (MacKenzie & Zhang, 1999; Soukoreff & MacKenzie, 1995). This could be achieved by statistically minimizing either the movement time (MT) or movement distance (D).

Let D_{ij} be the center to center distance from letter key i to letter key j on a given keyboard layout and P_{ij} be the frequency of the digraph letter j following letter i among all digraphs in a given language corpus (i.e., P_{ij} is the ratio between the number of $i-j$ digraphs and the total number of digraphs). One can calculate the average distance (d) traveled for tapping a character on that layout:

$$d = \sum_{i=1}^{26} \sum_{j=1}^{26} P_{ij} D_{ij} \qquad (1)$$

D_{ij} and d can be measured in any distance unit. Assuming all keys have the same size, an informative distance unit will be simply the key width (diameter if the key is round) so that the distance measure is normalized against key size and counted as the number of keys travelled.

Equation (1) is a reasonable but under-used optimization objective function. It means arranging the letter keys in such a way so the average movement distance is the lowest. For example, if the average travel distance to tap on a letter on QWERTY is 3.3 keys, a good result of optimization will be to lower that number to, for example, 1.5 keys. There are two advantages to such an optimization objective. First it is simple and direct, involving no models or parameters that may be subject to debate. Second it is also very meaningful: distance is literally and linearly related to "work" in physics terms. Minimizing the amount of work is a reasonable goal of optimization.

An alternative and more commonly used optimization metric is the average movement time (MT) taken to reach a key. Since movement time cannot be manipulated directly, it has to be related to layout in some way. One approach, taken for example by Hughes and colleagues (Hughes.D., Warren, & Buyukkokten, 2002), is to empirically measure T_{ij}, the average movement time between every pair of unlabeled keys on a grid. From there it was possible to obtain the average movement for a given layout with letters assigned on that grid.

A more common approach to movement time optimization uses a human movement model to compute movement time from distance. It is well known that movement time and travel distance are related in a simple equation known as Fitts' law (Fitts, 1954). According to Fitts' law, the time to move the tapping stylus from key i to key j for a given distance (D_{ij}) and key width (W_{ij}) is:

$$MT_{ij} = a + b \log_2 \left(\frac{D_{ij}}{W_{ij}} + 1 \right), \qquad (2)$$

where a and b are empirically determined coefficients. In other words, the more distant a key is from the movement starting point, or the smaller the key is in width, the longer the movement time will be.

It is possible to estimate the average text entry speed on a given touchscreen keyboard layout by summing up Fitts' law movement times between every pair of letters, weighted by the transitional frequency from one letter to another. Lewis and colleagues (Lewis, 1992; Lewis, Kennedy, et al., 1999; Lewis, Potosnak, & Magyar, 1997) were probably the first to use this calculation as a model of touchscreen keyboard performance. This model was more thoroughly and rigorously studied by MacKenzie and colleagues (MacKenzie & Zhang, 1999; Soukoreff & MacKenzie, 1995). Formally, it predicts the speed limit of tapping on a touchscreen keyboard as follows.

Let P_{ij} be the frequency of the ordered character pair i, j from N number of characters (typically but not necessarily 26 Roman letters); the mean time (t) for tapping a character is

$$t = \sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij} MT_{ij} \qquad (3)$$

Or,

$$t = a\sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij} + b\sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij} \log_2(\frac{D_{ij}}{W_{ij}} + 1) \qquad (4)$$

Since $\sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij} = 1$,

$$t = a + b\sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij} \log_2(\frac{D_{ij}}{W_{ij}} + 1) \qquad (5)$$

t has the unit of seconds. t can be converted to input speed (V) in characters per minute (CPM): $V = 60/t$. Equation (5) has been called the Fitts-digraph model (Zhai, Hunter, et al., 2002) .

There are two main advantages to use movement time as the objective function. First, time might be what users are most concerned about in entering text. Second, movement time as the objective function can be converted to the conventional typing speed units of characters per minute (CPM) or words per minute (WPM). For example if the conventional typing speed standard for an office worker on a typewriter is 60 WPM, an optimization result of 40 WPM for a touchscreen keyboard would give us a good idea how good the result is. From CPM to WPM is a simple arithmetic conversion given 1 minute = 60 seconds and 1 word = 5 characters. The latter is simply a convention and it includes the space character after each word. The average number of characters in a word depends on the text corpus. Our calculation from the American National Written Text Corpus is 4.7 characters per word excluding the space after the word. We choose to use CPM in the rest of this paper to avoid confusion.

On the other hand, there are disadvantages to using Fitts' law, primarily because there is a wide range of values of Fitts' law parameters (a and b in Equation 2) reported in the literature. Based on results from the more general Fitts' reciprocal tapping tasks, previous researchers have selected values such as a = 0, b = 0.204 (MacKenzie & Zhang, 1999; Zhai, et al., 2000). Specifically in the context of touchscreen keyboarding, more appropriate estimates were made at a = 0.083 s and b = 0.127 s (Zhai, Sue, et al., 2002). We employed these parameters in the present work. The accuracies of these values were subsequently verified by an empirical study to be reported later in this paper. If we assumed $a = 0$ as it had been mistakenly done in the literature (MacKenzie & Zhang, 1999; Zhai, et al., 2000), the percentage of time improvement estimated would tend to be exaggerated. a reflects "non-informational aspect of pointing action" (Zhai, 2004). Non-informational aspects of pointing here could include activation of muscles and key press actions that are independent of the distance to the targeted letter, hence not influenced by keyboard layout.

Fitts' law also suggests that movement time might be saved by giving greater size to the more commonly used letters. Although not a settled topic, previous work on key size optimization has been unsuccessful due to a number of reasons including the difficulty of tightly packing keys with varying size and the conflict between central positions and size (Zhai, Hunter, et al., 2002).

Given the pros and cons of each approach, the present work uses movement time as calculated in Equation 5 as the primary objective function, but we also simultaneously report the mean distance calculated according to Equation 1.

2.4 Optimization Scope

While the methods are general and to a large extent independent of the scope of keys included, we chose to first optimize only for the 26 Roman letters, excluding all auxiliary keys. Previous research in this area tended to include the space key in optimization because it is the most frequent character (MacKenzie & Zhang, 1999; Zhai, et al., 2000). The choice of excluding the space key in the current work is made for three reasons. First, on almost all the existing keyboards, including QWERTY and the keypad on a cell phone, the 26 Roman letters tend to be grouped together, while all other keys are arranged in the periphery. To ease the access of frequently used auxiliary keys, they are usually assigned different shapes and positioned at distinct positions (e.g., the spacebar on QWERTY). We keep this layout style when designing new keyboards to leverage users' familiarities and the possible cognitive advantages. Second, it is debatable what the best way to enter space is. Besides assigning the space key a distinct shape and position, previous research (Kristensson & Zhai, 2005) has argued that it is better to use a stroke gesture on the keyboard, such as a slide over more than one key-width distance to enter a space because such an action can be done anywhere on the keyboard (saving the time to travel to a specific location) and it is also a more robust word segmentation signal for word level error correction. In the case of gesture keyboards i.e. using stroke gestures approximately connecting letters on a keyboard as a way of entering words (Zhai & Kristensson, 2006), the space is automatically entered. Furthermore, the same optimization methodology used in the present work can also be applied if a space key is included in the optimization process. Including it will not change the essence and main conclusions of the present study.

Many languages use diacritics. It is possible to include letters with diacritics in the optimization process or to arrange them separately. We deal with this topic later in the paper.

To establish reference points, we first calculated the average tapping time and distance on two existing layouts (QWERTY and ATOMIK) as a touchscreen keyboard for English, using English digraph frequencies obtained from a modern large scale English corpus, the American National Corpus (ANC) containing 22,162,602 words.

Pervious research has shown that layout optimization is not sensitive to corpus source within the same language (Zhai, Hunter, et al., 2002).

By applying Equation 5, the English input speed $V_{English}$ of the QWERTY keyboard is estimated at 181.2 CPM. As another reference, the same model estimates the revised ATOMIK layout, available in the ShapeWriter gesture keyboard (Figure 2) as deployed in iPhone App Store in 2007, at 221.5 CPM. The revision of ATOMIK layout for ShapeWriter removed the space key from the center of the original ATOMIK (Zhai et al., 2002) for the purpose of shape writing. Applying Equation 1, we obtained the average travel distance per character on QWERTY and ShapeWriter ATOMIK at 3.31 and 1.94 key widths respectively.

Figure 2. A screenshot of the ATOMIK alternative layout embeded in the iPhone ShapeWriter software. It has an average travel distance of 1.94 keys per letter whereas the conventional QWERTY has an average of travel distance of 3.31 keys per letter

2.5 Optimization Methods and Constraints

Various approaches can be applied to optimize soft keyboard layouts. Borrowing from physical scientists' methods in understanding material or molecule structures as determined by the lowest energy state, Zhai, Hunter and Smith proposed to view the Fitts-digraph model as a "virtual energy" function and apply the Metropolis random walk algorithm to search for optimized touchscreen keyboards (Zhai, et al., 2000). They obtained the Metropolis/ATOMIK family of touchscreen keyboard layouts (Zhai, et al., 2000; Zhai, Hunter, et al., 2002) in this approach.

This approach consists of a finite number of iterations. In each iteration, the Metropolis algorithm picks up two random keys on the keyboard and swaps their positions to reach a new configuration. The input speed of a new configuration is then estimated based on the Fitts-digraph model (2). Whether the new configuration is kept as the starting position for the next iteration depends on the following Metropolis function:

$$\begin{aligned} W(O-N) &= e^{\frac{-\Delta t}{kT}} \; if \, \Delta t > 0 \\ &= 1 \qquad if \; \Delta t \le 0 \end{aligned} \qquad (6)$$

In Equation 6, $W(O-N)$ is the probability of changing from configuration O (old) to configuration N (new); $\Delta t = t_{new} - t_{old}$, where t_{new}, and t_{old} are mean times for typing a character on the new and old keyboard configurations, estimated by Equation (5); k is a coefficient; T is "temperature", which can be interactively adjusted. Key to the algorithm is the fact that the search does not always move toward a lower energy state. It occasionally allows moves with positive energy changes to be able to climb out of a local minimum. We use the same basic optimization process in the present work.

The conventional QWERTY keyboard lays the 26 English letters in a rectangle shape with 3 rows and 10 columns, which is well suited for two handed typing. Such a constraint is not necessary for touch screen keyboards. If the keyboard is not constrained to any particular shape, the objective function of minimizing one point travel (by a stylus or a single finger) would tend to result in a rounded keyboard (Zhai, et al., 2000). Our experience is that a more practical square shape (or near square such as a 5 rows by 6 columns grid) is more practical for graphic design and still gives sufficient flexibility for optimization (Zhai, Hunter, et al., 2002). As shown in Figure 2, in practical product designs the keys on the edge unused by the alphabet letters can be filled with auxiliary characters and functions.

3. OPTIMIZING KEYBOARDS FOR MULTIPLE LANGUAGES

3.1 Multilingual optimization methodology

The possible compatibility across different languages and the flexibility of the keyboard optimization space observed in the introduction give hope to the goal of simultaneously optimizing for multiple languages. The basic methodology used in the present study is to average the mean time of tapping a character across multiple languages, and minimize it by the Metropolis algorithm. Given m different languages, the mean time t of tapping a character across these m languages is calculated as:

$$t = \sum_{i=1}^{m} \frac{t_i}{m} \qquad (7)$$

t_i represents the mean time of inputting a character in language i, which is estimated by the Fitts-digraph model (Equation 5). t is then regarded as the objective function and minimized by the Metropolis algorithm. The optimization process is similar to previous work (Zhai, Hunter, et al., 2002), except that the "virtual energy" (t_{new} and t_{old}) is estimated by Equation 7.

The average in Equation 7 can be possibly weighted according to each language's speaker population or some other criteria. Although any weighting scheme can be controversial, the same methodology and procedure presented in this paper should still apply. We stay with Equation 7 as the objective function in the present work.

3.2 Optimization for English and French

We start with optimizing a keyboard for both English and French. These two languages are commonly used in the world: it is estimated that over 470 million people speak English and more than 250 million people speak French. The keyboard optimized for these two languages could benefit many English-French speakers.

As shown in Figure 3 and 4., although English and French vary in many aspects, their letter and digraph frequency distributions are strongly correlated, with correlation coefficients 0.88 (letter) and 0.75 (digraph) respectively. These high correlations are encouraging for our optimization objective. Note that the English corpus is obtained from ANC, and corpora of other languages are from http://corpus.leeds.ac.uk/list.html .

Figure 3. Letter Frequency of English and French

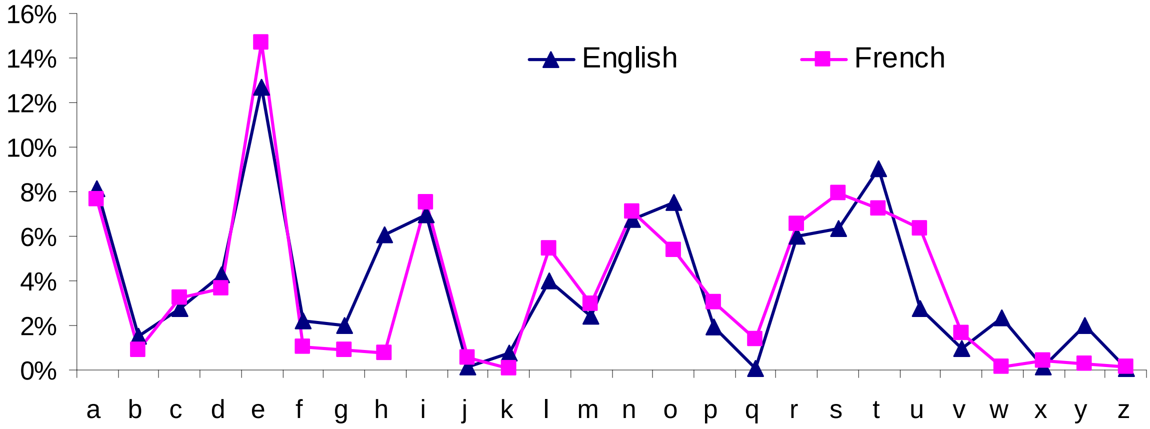
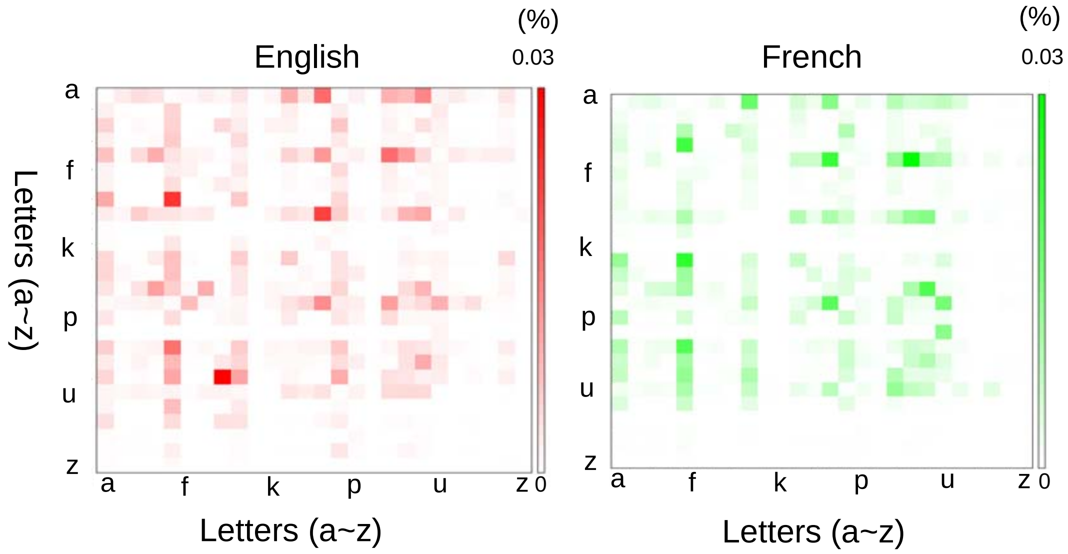


Figure 4. Heat map of English (left) and French (right) digraph distribution. The color intensity reflects each digraph cell's frequency



According to Equation 7, the mean time of typing a character t is then calculated as:

$$t = 0.5t_{Eng} + 0.5t_{Fren} \quad (8)$$

where t_{Eng} is the mean time of typing an English letter, and t_{Fren} a French letter.

Using the Metropolis algorithm, we obtained a variety of keyboards with similar performance. Figure 5 shows one of them, which is denoted by K-Eng-Fren.

Figure 5. The layout of K-Eng-Fren, K-English, and K-French

K-Eng-Fren

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

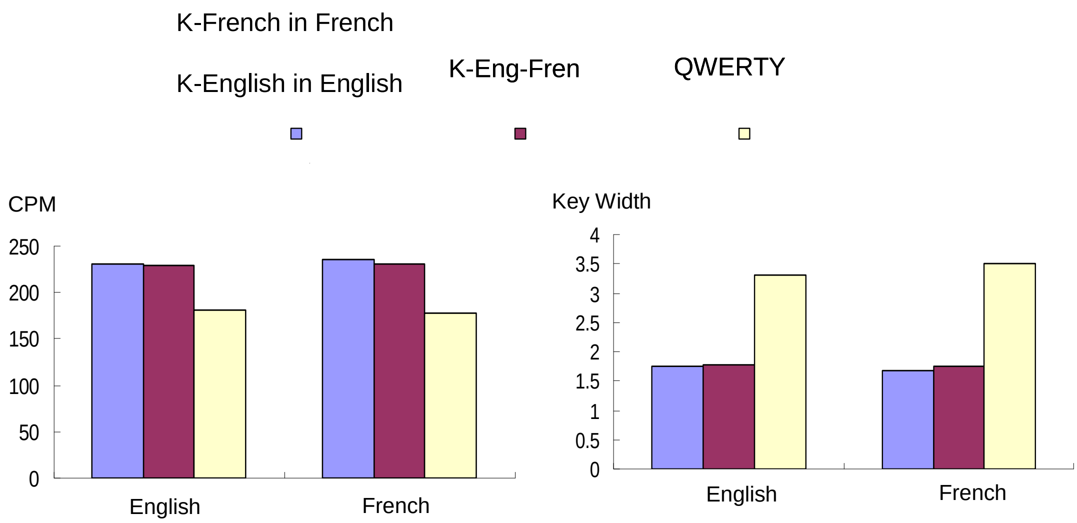
K-English

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

K-French

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

Figure 6. Predicted input speed and average travel distance per tap of K-English, K-French, and K-Eng-Fren and QWERTY



(a) Predicted Input Speed (b) Average Travel Distance Per Tap

We also used the Metropolis method to optimize solely for English and French, obtaining K-English and K-French respectively (Figure 5). Figure 6 shows the performance of all three optimization results, as well as the performance of QWERTY. The calculated English input speed for K-English, K-Eng-Fren, and QWERTY are 230.1, 229.4, and 181.2 CPM respectively, and French input speed for K-French, K-Eng-Fren and QWERTY are 235.0, 230.8, and 177.8 CPM respectively. As one would expect, since K-Eng-Fren takes into account two languages simultaneously, it is inferior to K-English in English inputting speed ($V_{English}$). However, K-Eng-Fren is only 1 CPM lower than K-English in speed. Such a tiny performance drop should be negligible in real use. A similar relationship exists between K-Eng-Fren and K-Fren. Compared to the standard QWERTY layout, K-Eng-Fren is superior in both English and French input: K-Eng-Fren improves the English inputting speed by 48.2 CPM, and French by 53.0 CPM.

If we look at the average travel distance per tap (Figure 6b) on these layouts for English and French respectively, we can draw similar conclusions. For English, the average travel distance per key press on K-English, K-French, and K-Eng-Fren and QWERTY are 1.76, 1.93, 1.78, 3.31 respectively. K-Eng-Fren is 46% shorter than QWERTY but only 1% longer than K-English. For French, the average travel distance per key press on K-English, K-French, and K-Eng-Fren and QWERTY are 1.85, 1.68, 1.76, 3.51 respectively. K-Eng-Fren is 50% shorter than QWERTY but only 5% longer than K-French.

In summary, it is somewhat surprising, and quite encouraging, that it is possible to simultaneously optimize a keyboard layout for both English and French input efficiency. The resulting layout has little loss for English from a keyboard specifically optimized for English and little loss for French from a keyboard specifically optimized for French.

3.3 Optimization for English, French, Spanish, German and Chinese

We have shown that it is possible to optimize movement time simultaneously for at least two languages, English and French, without a practical loss in efficiency for each. This is certainly good news for optimizing keyboard design between a pair of languages . Bilingual optimization is a possible strategy since most multilingual users are likely to be bilingual. However optimizing for two languages at a time can also be a problem due to the large number of combinations (e.g. English-French, English-Chinese, English-German, Chinese-French, Chinese-German, German-French …). The large number of combinations would impose configuration burdens on software and device makers, distributors, and users. Configuration simplicity can be one reason to maintain the status quo legacy of QWERTY for touchscreen keyboarding. This led us to tackle the next challenge – optimizing for five major languages, English, French, German, Spanish and Chinese (pinyin) at the same time.

We can attribute the positive results of K-Eng-Fren to the flexibility of the touchscreen keyboard optimization space and the high correlation in digraph distributions between these two languages. The question now is whether these same factors can still allow for simultaneously optimized keyboards for four large European languages plus Chinese *pinyin*. Spanish and German share some common words with English and French and all of them use the basic Latin alphabet. However, Chinese is different. It is logographic and shares few commonalities with the other four languages. Optimizing a keyboard for all of these five languages will help us understand how flexible the touchscreen keyboard optimization space is.

Moreover, these five languages are widely used in the world. Chinese, Spanish, and English are the top 3 most-spoken languages. The layout optimized for these five languages would benefit a large number of multilingual speakers.

Although Chinese is logographic, the most common approach to inputting Chinese on computers is based on *pinyin*, a phonetic spelling system also based on the Latin alphabet. Using *pinyin* to enter Chinese characters consists of two steps. First, the *pinyin* letters representing the pronunciation of the Chinese character is entered. As one *pinyin* string may correspond to many Chinese characters, the target character is then selected from the many homophonic candidates in the second step. Predictive techniques are usually used in this step: the ranking or even automatic selection of the corresponding logographic characters can be based on the proceeding characters in addition to the *pinyin* input. The first step involves typing letters, so keyboard layout could significantly affect its input speed. The second step is independent of the keyboard layout. To improve the input speed of Chinese via *pinyin,* keys should be arranged to improve *pinyin* input speed.

By analyzing language corpora, greater deviation was observed of Chinese from the other four languages (see Figure 7. for letter frequency). For example, the most frequent letter in English, French, Spanish, and German is “e”, while “i” is the most frequently used one in Chinese *pinyin*. Inconsistencies also exist in digraph distribution. For example, “zh” is frequently used in Chinese *pinyin*, while it rarely occurs in English.

Figure 7. Letter Frequency in English, French, German, Spanish and Chinese

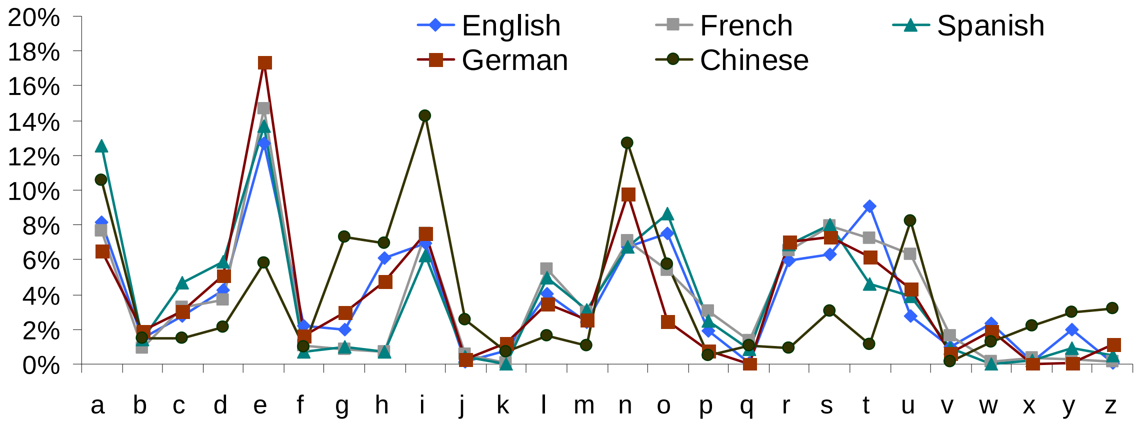


Table 1. Correlation coefficients of letter (bottom-left) and digraph frequency distributions (top-right)

	English	French	Spanish	German	Chinese
English		0.75	0.74	0.69	0.36
French	0.88		0.86	0.70	0.30
Spanish	0.86	0.92		0.66	0.24
German	0.88	0.90	0.81		0.33
Chinese	0.48	0.44	0.46	0.51	

Table 1 shows the correlation coefficients across these five languages. As shown, Chinese is loosely correlated with other four languages. In letter frequency distribution, the correlation coefficients between Chinese with other languages are all below 0.51, while those across other four languages are all above 0.8. In digraph distribution, correlation coefficients between Chinese with other four languages are all below 0.4. In one case, between Spanish and Chinese, the digraph correlation is only 0.24. These low correlations impose a challenge to simultaneous optimization. If the positive result in the last section was largely due to the strong digraph correlation between English and French, the current goal of optimizing for five languages including Chinese would have to place greater hope on the flexibility of the optimization space itself.

Following the proposed multilingual optimization methodology, each of the five languages is weighted equally in the optimization process. The mean time of inputting a character is represented as:

$$t = 0.2t_{Eng} + 0.2t_{Fren} + 0.2t_{Spanish} + 0.2t_{German} + 0.2t_{Chn} \qquad (9)$$

t_{Eng}, t_{Fren}, $t_{Spanish}$, t_{German} and t_{Chn} are the mean times of typing a character in the corresponding languages, which are predicted by the Fitts-digraph model (Equation 5).

By means of the Metropolis algorithm, we obtained the K5 layout which was simultaneously optimized for inputting English, French, Spanish, German, and Chinese. We also obtained optimized layouts and their performances for each of the five languages (See Figure 8). Table 2 summarizes these results in V (calculated input speed) and d (average distance in the unit of key width for entering a character) metrics.

Figure 8. The layouts of K5, and optimized layout for each individual language

K5

	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**

K-English

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

K-French

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

K-Spanish

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

K-German

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

K-Chinese

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

Table 2. Calculated input speed V (CPM) and average travel distance per tap d (keys) of various layout optimized for English, French, Spanish, German, Chinese, all five, and two previous layouts (ATOMIK and QWERTY)

	English		French		Spanish		German		Chinese	
	V	d	V	d	V	d	V	d	V	d
K-English	**230.7**	**1.76**	226.5	1.85	221.2	1.9	224.5	1.87	220.3	1.95
K-French	221.7	1.93	**235**	**1.68**	224.1	1.85	225.9	1.84	207.8	2.25
K-Spanish	217.4	2.02	229.2	1.78	**229.9**	**1.76**	223.9	1.89	201.3	2.37
K-German	221.5	1.94	221.2	1.95	218	1.97	**237.8**	**1.63**	211.9	2.12
K-Chinese	205.1	2.34	207.6	2.27	204.3	2.31	213.1	2.14	**244.9**	**1.5**
K5	**225.1**	**1.88**	**226.2**	**1.86**	**221.6**	**1.91**	**230.6**	**1.77**	**233.4**	**1.68**
ATOMIK	221.5	1.94	221.2	1.96	215.9	2.05	222	1.93	212.8	2.1
QWERTY	**181.2**	**3.31**	**177.8**	**3.51**	**173**	**3.7**	**181.9**	**3.26**	**168.7**	**3.85**

Let us first examine the individual results when optimized specifically for each of the five languages, summarized in bold numbers in the diagonal cells of Table 2. The first interesting observation is that after optimization for each language the average travel distance per tap all fell into a relatively narrow range: 1.76, 1.68, 1.76, 1.63 and 1.5 keys for English, French, Spanish, German and Chinese respectively. One might have expected greater differences between these languages given their different phonology. In comparison QWERTY is somewhat equally bad for all: 3.31, 3.51, 3.7, 3.26, 3.85 keys for English, French, Spanish, German and Chinese respectively. The ratio between the average travel distance per tap on QWERTY and the average travel distance per tap on the keyboards individually optimized for each language are large: 1.88, 2.09, 2.10, 1.99, 2.57 for English, French, Spanish, German, and Chinese respectively. Figure 8 illustrates the travel distance difference among the various layouts. Although English has been the primary target language used in touchscreen keyboard optimization work (Getschow, et al., 1986; Lewis, 1992; Lewis, Kennedy, et al., 1999; Lewis, et al., 1997; MacKenzie & Zhang, 1999; Soukoreff & MacKenzie, 1995; Zhai, et al., 2000; Zhai, Hunter, et al., 2002), English in fact has the least to gain and Chinese has the most to gain from touchscreen keyboard optimization. These results and observations are new to our knowledge.

When the five languages are considered simultaneously, the optimization effect is still strong. As shown in Table 2, the average distance d for tapping a character on K5 are 1.88, 1.86, 1.91, 1.77 and 1.68 keys for English, French, Spanish, German and Chinese respectively, much shorter than QWERTY's 3,31, 3.51, 3.7, 3.26, and 3.85 keys and close to the results obtained specifically for each language (1.76, 1.68, 1.76, 1.63 and 1.5 keys for English, French, Spanish, German and Chinese respectively.) The ratios in travel distance between K5 and QWERTY, and between the individually optimized layouts and K5 are summarized in Table 3.

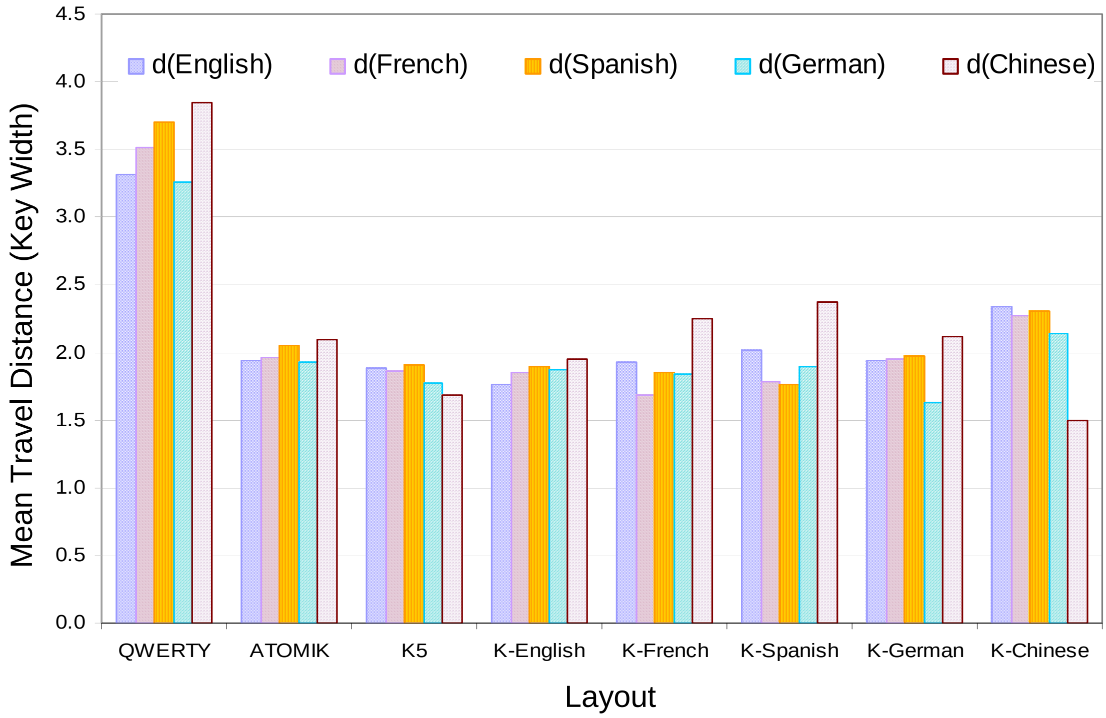
Table 3. The ratios of travel distance between layouts for the five languages

	English	French	Spanish	German	Chinese
Individual optimization/QWERTY	0.53	0.48	0.48	0.50	0.39
K5/QWERTY	0.57	0.53	0.52	0.54	0.44
Individual optimization/K5	0.94	0.90	0.92	0.92	0.89

As we discussed earlier, the digraph correlation among the five languages are relatively weak so by optimizing for only one language there would be no guarantee that the resulting layout would also be good for other languages. Note that the optimization process uses a stochastic method so each layout obtained is just one instance of many possibilities. The specific instance of layout we obtained for English happened to be also quite good for the other four languages (see Figure 9), although not as good as K5. On the other hand, the specific instance of Spanish layout was relatively poor for Chinese.

Interestingly, the layout optimized for Chinese was not very good for any of the other four languages (Figure 9).

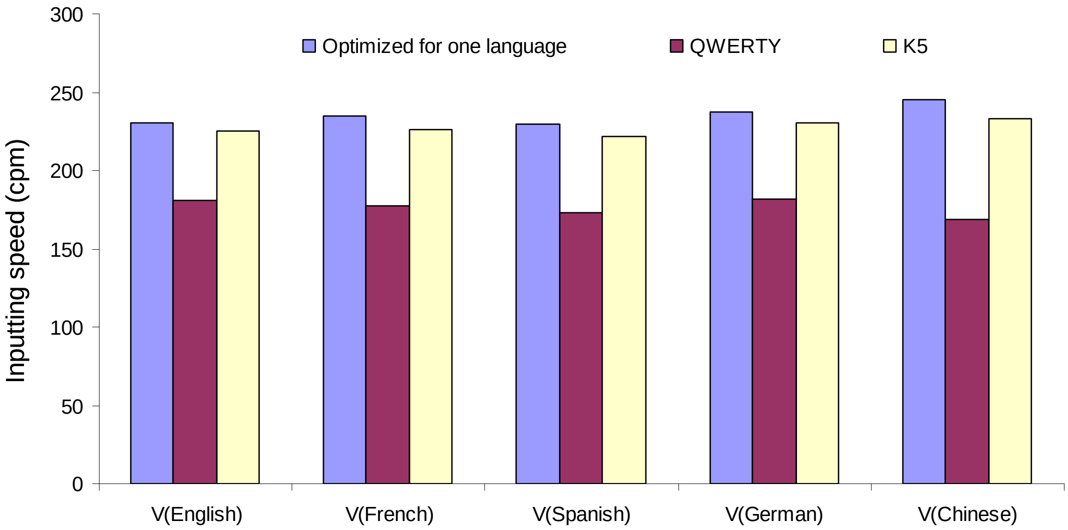
Figure 9. The average travel distance per tap for English, French, Spanish, German, and Chinese on various optimized and QWERTY layouts



The computational study and analysis thus far have not only produced optimized layouts for French, Spanish, German and Chinese that have not been previously reported in the literature, but also demonstrated that it is possible to accommodate at least these five languages in one optimized layout with about a 10% travel distance increase from individually optimized layouts (See Table 3).

Having examined the layouts in terms of travel distance, let's now evaluate the calculated input speeds of all languages as shown in Figure 10 and in Table 2. K5 is faster than QWERTY for all the five languages. Take English as an example, K5 improves the input speed by 24% over QWERTY, from 181.2 CPM to 225.1 CPM. To compare with a past optimization effort, the performance metrics (*V and d*) of K5 for English are better than those of the revised ATOMIK as currently used in ShapeWriter (V = 221.5 CPM, d = 1.94 keys).

Figure 10 Calculated input speed of K5, QWERTY, and one language optimization keyboards



Fortunately, considering five languages simultaneously caused only minimal performance decreases to the optimization results: the input speed of K5 is very close to K-English, K-French, K-Spanish, K-German and K-Chinese, separately optimized for each language. The biggest decrease occurs in Chinese entry, in which K5 is 11.5 CPM, or around 5%, slower than K-Chinese.

The five popular languages investigated in this paper are from diverse groups: English and German are Germanic languages, and Spanish and French are from the Romance language group. Although all four of these languages belong to the Indo-European family, Chinese is from the Sino-Tibetan language family. Despite the diversity, our result shows that there are near optimal layouts that can accommodate all of them, demonstrating the flexibility of the touch screen optimization space. While these five languages together are spoken by nearly two thirds of the world population, they are a small fraction of the world total number of languages. Furthermore all of these five investigated, including Chinese *pinyin*, are based on the Latin alphabet. How much more we may expand this common optimization approach to cover even more languages, particularly those that are not Latin alphabet-based, remains an open question.

4. EMPIRICAL VERIFICATION

The main questions in the present work are theoretical and mathematical, not empirical. The mathematical and computational work presented so far has lended support to a somewhat counterintuitive conjecture: it is possible to optimize a touchscreen keyboard simultaneously for multiple languages to the extent that the resulting layout is nearly as efficient as individually optimized keyboards for each of the multiple languages. In particular, K5, a keyboard simultaneously optimized for five languages, is superior to the standard QWERTY keyboard for English, French, German, Spanish and Chinese input with significant reduction in average travel distance in all five.

The reduction in average travel distances is a numeric fact and needs no empirical test. But, can users actually achieve the tapping speed predicted by the Fitts-digraph model with sufficient practice?

The last question has two components. The first involves learning and the second has to do with the Fitts' law relationship between distance and time. Recall that the Fitts-digraph model assumes that after sufficient practice, users of an optimized touchscreen keyboard would learn the new layout so well that their input speed would be primarily limited by movement control rather than visual search. This assumption has been repeatedly validated on different optimized layouts. It has been shown that experimental participants could indeed reach Fitts-digraph model predictions in English (MacKenzie & Zhang, 1999; Zhai, Sue, et al., 2002) and there no reason to expect that movement time from key to key will be language dependent. In other words, it is not necessary to run text entry learning experiments in the current context.

On the other hand, although Fitts' law has been tested for decades and the Fitts-digraph model has been empirically tested in the context of stylus keyboarding previously, we felt that it would be prudent to test the model validity under the specific parameter choices made for the present work. It would also be reassuring to see how well empirical data match the model predictions even for a small set of well learned words. This consideration led us to conduct an experiment verifying the Fitts-digraph prediction on the K5 layout with a set of sample English words. Since all five languages were equally treated in the touchscreen keyboard design process, any language is eligible for the test. We chose English in this experiment.

We included the optimized K5 and the conventional QWERTY layout in the test. These two layouts should be sufficient for verifying the Fitts-digraph model and its parameters in this paper. Once verified, the model established can be applied to other layouts with confidence.

4.1 Method

Apparatus

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 experiment software was developed in JAVA and ran on Window XP. The size of each key was 1 cm x 1 cm.

Participants

Twelve volunteers (4 female, 8 male), 26 to 55 years old, participated in the experiment. All were right-handed. Nine participants used a touch-screen device at least once per day, and others about once a month.

Tasks

A repeated tapping task was designed to verify the Fitts-digraph model which indicates the input speed of an average user who has learned a given layout to the stage when performance is limited by movement constraints, not visual or cognitive components (MacKenzie & Zhang, 1999; Soukoreff & MacKenzie, 1995). A set of 19

English words were tested in random order. Participants were asked to tap each word repeatedly ten times in a row (Trial T1 to T10) to reach the expert input speed. In case of a mistake, the user had to click a button to clear it and repeat the word. The 19 words were:

the and you that is in of know not they get have were are bit quick fox jumps lazy

Adopted from (Zhai & Kristensson, 2008), these words cover all letters in English so together they touch on every letter key in a keyboard. Furthermore they have a high correlation in letter frequency with the spoken American National Corpus (R2 = 0.88) so they are representative in letter coverage. Note again the purpose of this experiment was not to demonstrate users' ability to learn an optimized layout for realistic typing tasks, which had been demonstrated before. The purpose was rather to test the Fitts-digraph model's precision, particularly as applied to the new K5 layout and the traditional QWERTY layout. For this purpose, these 19 words should give sufficient and diverse test samples since these words involve very different letter transitions.

Design

A within-subjects design with repeated measures was employed. The independent variables were two touchscreen keyboard layouts (K5 and QWERTY) and 19 English words. Each participant performed the repeated tapping tasks on both of the two layouts, with order of appearance 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 as possible and as accurately as possible. Breaks were enforced between changes of touchscreen keyboard layouts.

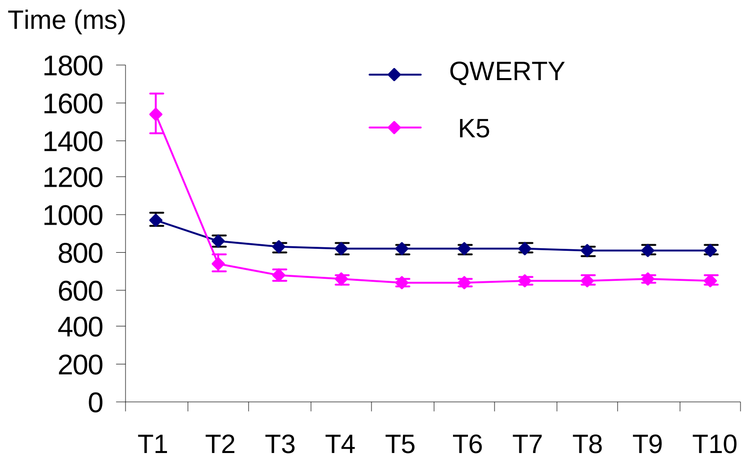
Measures

The dependent variable is *completion time,* defined as the time from the moment the first letter of the targeted word is tapped until the last letter of the targeted word is tapped. We focused on the interval between the first and the last letter to verify the Fitts-digraph model used in our optimization process.

4.2 Results

The percentage of erroneous trials was between 0.3% and 4% for any word on any layout. The mean word completion time on each keyboard is shown in Figure 11.

Figure 11. Mean completion time (Std. Error) per word

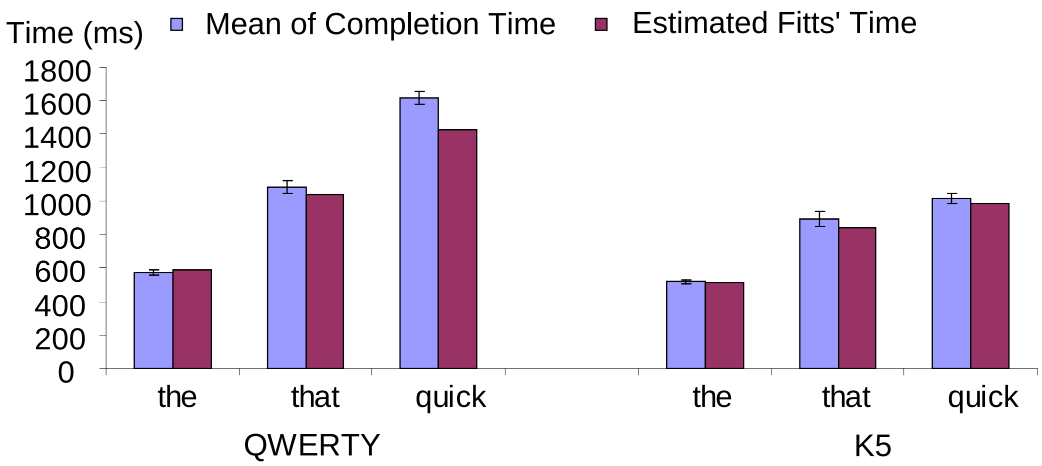


For both layouts, the mean completion time dropped quickly from trial T1 to T2, and reached a performance plateau after T3. Not surprisingly due to people's familiarity with QWERTY, the participants were much faster on it in the first trial but in subsequent repeated trials they were faster on the optimized layout K5.

We used data from T6 to T10 to reflect users' eventual expert input speed that is limited by Fitts' law movement constraints. Within these trials, repeated measure variance analysis showed a significant main effect on completion time for layout ($F_{1,11}$ = 54.498, p<.0001). The mean completion time per word was 813.457 ms (SD = 388.462 ms) for QWERTY, and 648.084ms (SD = 339.461ms) for K5. No significant main effect was observed on completion time for trials from T6 to T10 ($F_{4,44}$ = 0.535, p = .711), indicating that the participants entered words with similar speed from T6 to T10.

For both layouts, the mean completion time of a word in the stabilized trials (from T6 to T10) highly correlated with its Fitts' law predictions. For the 19 words tested, we computed their theoretical tapping times according to Fitts' law (Equation 2) with its parameter set at a = 0.083 s and b = 0.127 s (from Zhai, Sue, et al., 2002). The correlation coefficients between the measured mean completion time and corresponding Fitts' law prediction were 0.984 for QWERTY, and 0.995 for K5, respectively. Figure 12 shows the results for the words "the", "that", and "quick", representing three-letter, four-letter, and five-letter long words respectively. As we can see, the mean completion times are very close to the predicted completion time. This high correlation strongly validates the Fitts-digraph model: we indeed can rely on it to predict touchscreen keyboard performance.

Figure 12. Mean and std. error of completion time, and Fitts' time for the words "the", "that", and "quick".



4.3 Discussion

The high correlation between the model predictions and the empirical mean completion times of the words tested, and the close match between the absolute values of the predictions and empirical means strongly validated both the Fitts-digraph model and its parameters chosen in the current study. Although the tests were done in English, there were no intrinsic differences between tapping movement between different languages (save the accented letter issues to be addressed in the next section). The Fitts-digraph model and its predictions are therefore likely to be valid for the other four languages. All these results provide additional empirical assurance to the computational time estimates based on Fitts' law and its parameters a = 0.083 s and b = 0.127 s for touchscreen keyboard tapping (Zhai, Sue, et al., 2002).

5. INPUTTING DIACRITIC CHARACTERS

In addition to the 26 characters from the basic Latin alphabet, French, Spanish, and German scripts also contain diacritics marks and characters that do not exist in English, such as è, ç in French, and ñ in Spanish. However, the frequency of occurrence of these *diacritic characters* is less than 4% in each of these three languages (http://en.wikipedia.org/wiki/Letter_frequency).

There are two widely used approaches to entering *diacritic characters* in physical keyboards:

Direct-inclusion. This approach directly includes characters with diacritic marks, each as a separate key on the keyboard. For example, the AZERTY keyboard, a keyboard designed for inputting French, contains keys é and ç. The user can access them by directly pressing the key.

Dead-key. The second approach is to assign each diacritic mark with a dead key . A dead key is a special key on the keyboard which does not generate a character when struck, but allows modification on the following letter. The user types the base character after striking the dead key to input an accented letter. For example, the key strokes ~ and n result in the character ñ.

Dead-key is less efficient than *Direct-inclusion* if considered in isolation since it requires two key strokes for inputting a complete character. However, since the diacritic characters are low (less than 4% for each language) in letter distributions, the performance decrease caused by a dead-key approach might be negligible.

On the other hand, on touchscreen keyboards *dead-key* methods occupy less display space, saving valuable screen real estate for other keys and reducing the time needed to visually locate target keys.

To gain deeper insights into these two approaches on touchscreen keyboards, we optimized two multilingual touchscreen keyboard layouts that included means to enter all *diacritic characters* in French, Spanish, German, and Chinese *pinyin*. Similar to designing K5, the mean time of tapping a character is minimized by the Metropolis algorithm (Equation 6). K5-Direct is the layout on which *diacritic characters* are entered by direct-inclusion, while K5-Deadkeys is the one that employs dead-keys to input diacritic characters (Figure 13.). Both K5-Direct and K5-Deadkeys are optimized simultaneously for English, French, Spanish, German, and Chinese.

Figure 13. K5-Direct and K5-Deadkeys

K5-Direct

ñ	q	g	w	x	í	ë
y	u	n	t	c	j	â
f	o	a	i	h	z	à
b	l	r	e	s	ó	á
ü	p	m	d	v	ä	ú
ß	k	é	ê	û	ù	ç
î	ÿ	ô	è	ö	ò	ï

K5-Deadkeys

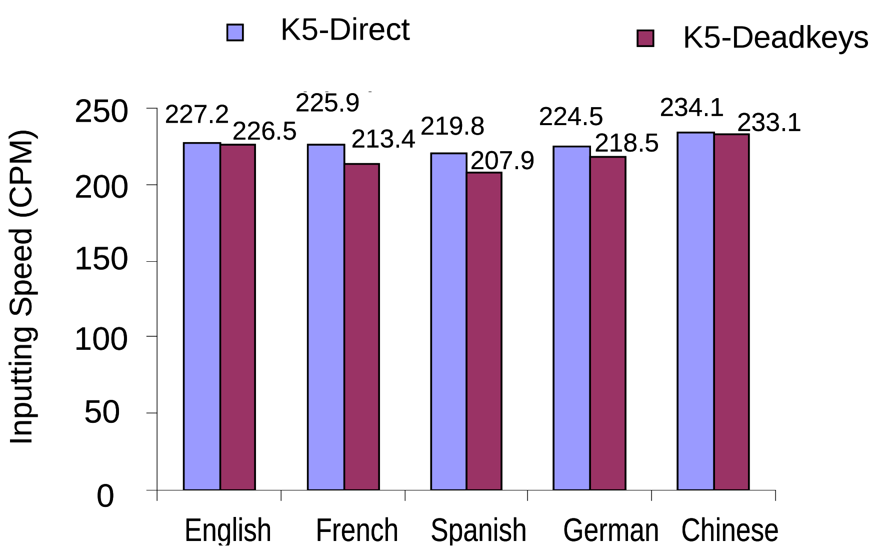
	¨	g	y	j	^
q	u	n	t	c	x
f	o	a	i	h	z
b	m	r	e	s	w
	p	l	d	´	¸
	ß	k	v	`	~

Note that since the letter ß can only be entered directly, we treat it as an individual letter in K5-Deadkeys. Similarly, ñ and ç could be treated as individual letters for these five languages.

As expected, K5-Direct is faster than K5-Deadkeys in inputting French, Spanish and German (Figure 14.), because the *Dead-Key* approach requires two key strokes for inputting a diacritic character. Surprisingly, K5-Direct is also slightly superior to K5-Deadkeys in inputting English and Chinese *pinyin*, although neither of these two languages requires diacritic characters input. We suspect that this is because English letters are arranged to accommodate diacritics input on the K5-Deadkeys layout (e.g., e is positioned close to ´ for inputting é). With this diacritics input constraint, the optimization space of the 26 English letters on K5-Deadkeys is smaller than that on K5-Direct, thus leading to slower inputting speed. Since the diacritics letters occur with low frequencies across all the five languages, considering them in optimization process does not cause big performance decreases. The biggest inputting speed difference between K5 and K5-Direct is in German input: the K5 is only 6.1 CPM faster than K5-Direct. As expected, average inputting speed of K5 across the five languages is faster than that of K5-Direct, albeit K5 is slightly slower than K5-Direct in English and Chinese input.

Although K5-Direct is faster than K5-Deadkeys across all the five languages, the differences are quite small. The biggest performance difference between K5-Direct and K5-Deadkeys is in French input, in which K5-Deadkeys is 10 CPM, or 5%, slower than K5-Direct. On the other hand, K5-Deadkeys saves 16 extra key spaces over K5-Direct, or 33% of the keys on K5-Direct.

Figure 14. Estimated input speed of K5-Direct and K5-Deadkeys

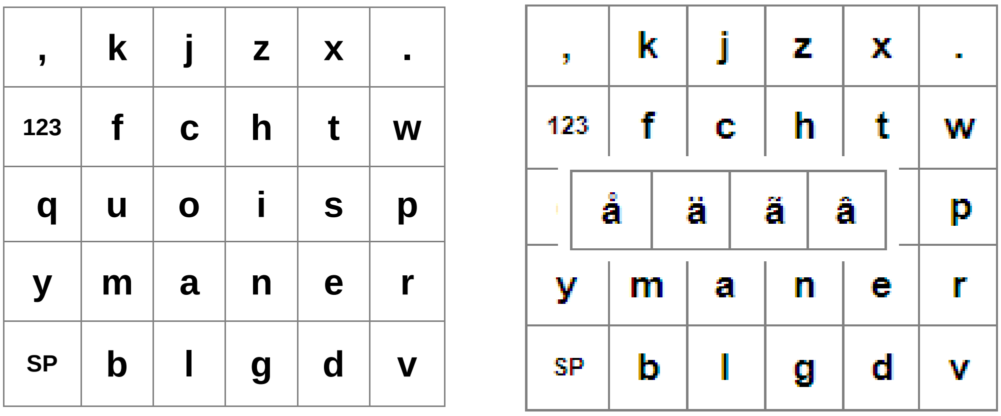


Incidentally, on either K5-Direct or K5-Deadkeys, 26 English letters are grouped together and surrounded by the non-English letters. This might be explained by the distribution of letter frequencies. The 26 English letters occur more frequently than the diacritic characters across all the languages, hence the optimization process tends to tie them together, resulting in short tapping distances between them. This arrangement style coincidently concurs with a generally employed touchscreen keyboard design strategy (Zhai et al., 2002) which treats the 26 English letters as a structural whole and surrounds them with auxiliary keys.

In conclusion, there is a trade-off between the dead-key and the direct-inclusion approaches. The dead key approach saves screen space and the direct-inclusion approach reduces the average input time albeit by only a small amount (between 0.3% and 5% depending on the language). Given that touchscreen devices are often mobile (with small screens); the dead-key approach might be preferable.

Yet another approach, "press-to-expand", is to multiplex the regular letter keys. For example, with a "long press" on the *a* key, additional letters with the same base *å ä ã â* appears as a menu for selection and each of them can be selected by an additional tap action (Figure 15). This approach, already used in commercial applications such as the iPhone touchscreen keyboard, is even more compact in space but should be even slower than the dead-key approach because not only it takes two taps to enter a diacritic letter, the first tap has to be a long press with a duration longer than a set threshold (e.g. 250 to 500 ms). However, again due to the low frequency of diacritic letters, their impact on the average input speed should still be relatively small. Another potentially useful approach is to add menu-augmentation to the keys so a diacritic letter can be entered by a sliding stroke from its base letter to one of the menu sections overlaid around the key (Isokoski, 2004).

Figure 15. The "Press-and-expand" method of entering diacritic characters: The default keyboard only displays regular keys (left). When a character (e.g. "a") is pressed and held for a set period of time, its diacritic variations are displayed for selection (right)



In short, with these understandings, the treatment of *diacritic characters* in multilingual touchscreen keyboards is a choice to be made depending on the device characteristics such as available space. Including these letters in the optimization process can but only slightly increase the average input speed due to their low frequency. However, efficiency may not be the only factor to be considered in design. Product designers should also consider how native speakers conceptualize diacritic characters. The main contribution of this section is the understanding of the quantitative influence of each type of design.

Other auxiliary keys such as punctuation and numeric keys are usually placed surrounding the primary characters. For a few most frequent keys (e.g. Space, and Backspace), they can be offered distinct appearances to become the "landmarks" among the other keys, thus helping users to locate them quickly. Product designers can arrange these keys on a touchscreen keyboard in consideration of visual appearance. Often on mobile devices they are arranged on an alternative panel to save space. Figure 2 shows an example of a touch screen keyboard design in practice.

Besides touchscreen keyboards, text can also be entered through handwriting on touchscreens with a pen or finger. The drawback of handwriting is the lack of efficiency.

Yet another approach is gesture keyboards that combine stroke gestures with a keyboard. As demonstrated in the SHARK/ShapeWriter project (Zhai & Kristensson, 2003), gesture keyboard defines a shorthand system which can be much more efficient.

6. CONCLUSIONS AND FUTURE WORK

Although a sizable amount of work has been undertaken in search of efficient touchscreen keyboards, the scope and nature of the current work is unprecedented. Built upon the past experience accumulated in the HCI research literature, the present work explores the optimality issues in multilingual touchscreen keyboard design, resulting in following contributions.

First and foremost, the work lends support to the conjecture that it is possible to simultaneously optimize a touchscreen keyboard for multiple languages at the same time, with only a small comprise for each language's input optimality. Before the current investigation, it was not known this was possible because one would imagine that optimizing for one language would be at a large cost for another. Once demonstrated possible, there could be many similar designs in the future.

Second, we proved the effectiveness of applying a set of methods based on the Fitts-digraph model and the Metropolis algorithm to tackle multilingual touchscreen keyboard optimization. The methods developed in the current work can be applied to other multiple languages with minor modifications.

Third, the work has produced a set of optimized layouts specifically for French, Spanish, German, and Chinese *pinyin* (Figure 8), as well as an English-French bilingual layout (Figure 5).

Fourth, the work has produced K5 (Figure 8) as well as K5-Direct and K5-Deadkeys (Figure 13.) that are simultaneously optimized for five commonly used languages: English, French, Spanish, German and Chinese. K5 reduces the average movement distance to 1.76, 1.68, 1.76, 1.63 and 1.5 keys from QWERTY's 3,31, 3.51, 3.7, 3.26, and 3.85 keys for English, French, Spanish, German and Chinese pinyin respectively. Computational analyses, which are reinforced by a subsequent empirical verification, show that K5 could significantly improve the input speed of each language in comparison to QWERTY. Compared to the keyboards separately optimized for each language, K5 only has minor performance decreases.

Finally, we investigated approaches for inputting diacritic characters in optimized multilingual touchscreen keyboards. The quantitative results show that having these diacritic characters participate in the multilingual keyboard optimization along with regular characters is beneficial to the average input time, but only slightly. The choice of direct-inclusion, dead-key, or key-multiplexing approaches can be left as product designers choice depending on touch space availability.

Although the five languages studied in the current work represent a very large population of potential users, one would still want to ask whether the result of such optimization can be further extended towards a "universal layout" and at what rate the individual language optimality will begin to be significantly sacrificed for collective optimality. Another interesting research direction is to optimize the layout for gesture keyboards originated in the SHARK/ShapeWriter project (Kristensson & Zhai, 2004; Zhai & Kristensson, 2003; Zhai & Kristensson, 2006). Additionally, many input methods now use predictive techniques. The target word could be selected from a candidate list after typing its prefix. It would be interesting to investigate whether and how these predictive techniques will affect the touch-screen keyboard optimization space in future work.

7. ACKNOWLEDGEMENT

The initial work of this paper was done when Xiaojun Bi worked as a graduate intern under Shumin Zhai’s supervision at the IBM Almaden Research Center. We thank the support of many IBM colleagues. We are particularly indebted to the action editor and the three exceedingly insightful reviewers of HCI whose comments and suggestions have markedly improved the paper.

REFERENCES

del Galdo, E. M., & Nielsen, J. (Eds.). (1996). *International users interface*. New York: John Wiley & Sons.

Dell'Amico, M., Díaz, J. C. D., Iori, M., & Montanari, R. (2009). The single-finger keyboard layout problem. *Computers & Operations Research 36*(11), 3002-3012.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology, 47,* 381-391.

Getschow, C. O., Rosen, M. J., & Goodenough-Trepagnier, C. (1986). *A systematic approach to design a minimum distance alphabetical keyboard.* Proceedings of RESNA (Rehabilitation Engineering Society of North America) 9th Annual Conference, Minneapolis, Minnesota, pp. 396-398.

Gordon, R. G., & Grimes, B. F. (Eds.). (2005). *Ethnologue: languages of the world* (15th ed.). Dallas: SIL International.

Hughes.D., Warren, J., & Buyukkokten, O. (2002). Empirical bi-action tables: a tool for the evaluation and optimization of text-input systems, applications I; stylus keyboard. *Human-Computer Interaction, 17*(2,3), 271-309.

Isokoski, P. (2004). *Performance of menu-augmented soft keyboards.* Proceedings of ACM CHI Conference on Human Factors in Computing Systems, pp. 423-430.

Kristensson, P.-O., & Zhai, S. (2004, Oct 24-27, 2004). *SHARK2: A Large Vocabulary Shorthand Writing System for Pen-based Computers.* Proceedings of ACM Symposium on User Interface Software and Technology (UIST 2004).

Kristensson, P. O., & Zhai, S. (2005). *Relaxing stylus typing precision by geometric pattern matching.* Proceedings of ACM International Conference on Intelligent User Interfaces, pp. 151-158.

Lewis, J. R. (1992). *Typing-key layouts for single-finger or stylus input: initial user preference and performance* (Technical Report No. 54729). Boca Raton, FL: International Business Machines Corporation.

Lewis, J. R., Kennedy, P. J., & LaLomia, M. J. (1999). *Development of a Digram-Based Typing Key Layout for Single-Finger/Stylus Input.* Proceedings of The Human Factors and Ergonomics Society 43rd Annual Meeting.

Lewis, J. R., LaLomia, M. J., & Kennedy, P. J. (1999). *Evaluation of Typing Key Layouts for Stylus Input.* Proceedings of The Human Factors and Ergonomics Society 43rd Annual Meeting.

Lewis, J. R., Potosnak, K. M., & Magyar, R. L. (1997). Keys and Keyboards. In M. G. Helander, T. K. Landauer & P. V. Prabhu (Eds.), *Handbook of human-computer interaction* (2nd ed., pp. 1285-1315). Amsterdam: Elsevier Science.

MacKenzie, I. S., & Tanaka-Ishii, K. (Eds.). (2007). *Text Entry Systems: Mobility, Accessibility, Universality*: Morgan Kaufmann Publishers.

MacKenzie, I. S., & Zhang, S. X. (1999). *The design and evaluation of a high-performance soft keyboard.* Proceedings of CHI'99: ACM Conference on Human Factors in Computing Systems, pp. 25-31.

Shneiderman, B. (2000). Universal usability. *Communications of the ACM. 43*(5), 84 - 91

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

Wang, J., Zhai, S., & Su, H. (2001). *Chinese Input with Keyboard and Eye Tracking - An Anatomical Study.* Proceedings of ACM CHI Conference on Human Factors in Computing Systems, pp. 349-356.

Yamada, H. (1980). A historical study of typewriters and typing methods: from the position of planning Japanese parallels. *Journal of Information Processing, 2*(4), 175-202.

Zhai, S. (2004). Characterizing computer input with Fitts' law parameters -- The information and non-information aspects of pointing. *International Journal of Human-Computer Studies*(Special Issue of Fitts (1954) 50th Anniversary).

Zhai, S., Hunter, M., & Smith, B. A. (2000). *The Metropolis Keyboard - an exploration of quantitative techniques for virtual keyboard design.* Proceedings of The 13th Annual ACM Symposium on User Interface Software and Technology (UIST), San Diego, California, pp. 119-218.

Zhai, S., Hunter, M., & Smith, B. A. (2002). Performance optimization of virtual keyboards. *Human-Computer Interaction, 17*(2,3), 89-129.

Zhai, S., & Kristensson, P.-O. (2003). *Shorthand Writing on Stylus Keyboard.* Proceedings of CHI 2003, ACM Conference on Human Factors in Computing Systems, CHI Letters 5(1), Fort Lauderdale, Florida, pp. 97-104.

Zhai, S., & Kristensson, P. O. (2006). *Introduction to Shape Writing*: IBM Research Report RJ10393 (also as a Chapter 7 of I. S. MacKenzie and K. Tanaka-Ishii (eds), Text Entry Systems: Mobility, Accessibility, Universality, Morgan Kaufmann Publishers, pp 139-158.

Zhai, S., & Kristensson, P. O. (2008). *Interlaced QWERTY: accommodating ease of visual search and input flexibility in shape writing.* Proceedings of ACM Conference on Human Factors in Computing Systems, pp. 593-596

Zhai, S., Sue, A., & Accot, J. (2002). *Movement model, hits distribution and learning in virtual Keyboarding.* Proceedings of CHI 2002: ACM Conference on Human Factors in Computing Systems, CHI Letters 4(1), Minneapolis, Minnesota, pp. 17-24.