Toward an Optimized Arabic Keyboard Design for Single-Pointer Applications

Abir Benabid Najjar
College of Computer and Information Sciences
King Saud University, Riyadh, Saudi Arabia
abbenabid@ksu.edu.sa

ABSTRACT

This paper introduces an ongoing project[1] that aims to design an Arabic keyboard for applications that predominantly use single pointer input device. Such applications are available in mobile devices like Portable Data Assistant (PDA) and Smart phones, as well as in gaze controlled interfaces which constitute an on-growing mode of communication, especially for people with mobility impairment. In this paper, we focus on the optimization of the key arrangement based on the movement time of one pointer (finger, stylus, eye...) and character transition frequencies in the Arabic language. Experimental results show that the optimized layout improves the overall typing speed and outperforms the commonly used Arabic keyboards.

Categories and Subject Descriptors

G.1.6 [**Optimization**]: Simulated annealing; H.5.2 [**User Interfaces**]: Input devices and strategies

Keywords

keyboard design, single pointer, eye typing, mobile devices, Quadratic Assignment Problem, metaheuristics.

1. INTRODUCTION

With the continuous evolution of mobile devices such as smart phones and PDA, they are increasingly used in writing long messages, emails and even text documents. The main interface for such applications is the virtual keyboard for which only one-pointer (finger or stylus) is available for data input. This type of virtual keyboards, with one-pointer based text entry, may be also used for applications that predominantly use eye typing interfaces for severely disabled people. These applications involve an eye tracking device that detects eye movements and an on-screen keyboard where the selection is computed within a predefined Dwell time that corresponds to a prolonged gaze fixation.

Several optimized keyboards have been proposed in the literature to enhance the typing speed for single finger or stylus-based text entry: Metropolis, OPTI, FITALY and YLAROF [4] keyboards attempt to minimize distances of the commonly associated pairs of characters in the English language.

Arabic keyboard

Arabic is one of the most populous languages in the world and is widely taught in schools and universities and used in workplaces and media. It uses the Arabic alphabet that has a particular script and it is written from right-to-left which affects the placement of characters in the keyboard. However, few optimized layouts for Arabic keyboards [6, 7] were reported in the literature, and none of these models considered the one pointer use. And we cannot rely on the results of research done on multiple-fingers keyboard layout, since these keyboards are optimized depending on hand alterations, the weight associated to each finger and the efforts distribution among the two hands.

Recently, an Arabic virtual keyboard, named iWriter[1], has been proposed for use by people with physical disabilities using gaze controlled text tapping. But the keyboard's layout used in this system is not optimized for eye typing. Thus, the authors highlighted the importance of designing an optimized Arabic keyboard layout adapted for such applications in order to improve as much as possible the text entry performance since (1) eye typing can be very slow due to the dwell time threshold that sets a limit on the maximum typing speed, (2) it can also led to eye fatigue and pain when involved with long time computer related work due to the repetitiveness of eye movements.

2. PROBLEM FORMULATION

For several years, the problem of keyboard design has been addressed by researchers in the Ergonomics domain before being formulated as a combinatorial optimization problem. The single-pointer keyboard layout problem can be expressed as a Quadratic Assignment Problem (QAP). A QAP is basically about assigning N facilities to N different locations while the corresponding cost depends on the distances between locations as well as the flows between facilities. To map the single-pointer keyboard layout problem to the QAP, we consider two sets:

- *The characters transition frequencies* $\{f_{ij}\}_{1\leq i,j\leq N}$ in Arabic language represent the flows. We used the transition frequencies between the different pairs of characters computed in [6] by parsing Arabic Wikipedia articles from different fields. These articles cover subjects across almost all disciplines and then the vocabulary is not reduced to a specific domain.

[1]This research project is supported in part by a grant (RC121241) from the Faculty of Computer Science and Information Research Center - King Saud University.

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands.
ACM 978-1-4503-1964-5/13/07.

- *The movement times* $\{T_{ij}\}_{1\leq i,j\leq N}$, needed to type two consecutive characters, represent the distances between locations. The movement time from one key to another may be estimated proportionally to the rectilinear distance between these two keys. It has been predominantly computed using Fitts law [3] which states that the time and difficulty, required to type consecutively two symbols i and j, can be estimated by

$$T_{ij} = a + b \cdot log_2(D_{ij}/A_j + 1) \quad (1)$$

D_{ij} is the distance of the two keys assigned to i and j, and A_j is the width of the target key assigned to j. We set the constant values a and b to respectively 0 and 1/4.9 bits/sec that have been experimentally determined by MacKenzie et al. [5] in the special case of using a stylus for data entry. We assume that the keys have equal widths which implies that $A_j = 1$. And then the movement time only depends on the distance of the two keys.

Then, for a given feasible solution S, we can compute the corresponding fitness function $\omega(S)$:

$$\omega(S) = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij} \cdot T_{ij} \quad (2)$$

This fitness function reflects the performance of a given keyboard arrangement and then would be used to evaluate the different keyboard's layouts.

3. OPTIMIZATION METHOD

The QAP is known to be an NP-hard problem [2] for which exact approaches are limited to small instances of the problem. Simulated Annealing (SA) is one of the metaheuristic approaches that have been applied with success for the QAP.

We considered a two stages SA approach to solve the corresponding single-pointer keyboard layout problem: The first stage starts with a randomly generated solution, then performs a greedy search trying to find a better solution by exchanging the position of any two characters until an improved layout can not be found in 10 consecutive attempts. The second stage corresponds to the SA search that starts from an initial high temperature T_0, so that many solutions with higher fitness functions are accepted. Then, we reduce the current temperature every L_{max} iterations by means of a linear reduction factor r.

After extensive experiments, we set the initial temperature T_0 to 10^7, the maximum number of iterations, at the same temperature, L_{max} to 15, and the maximum number of reheating steps M_{max} to 10, with a factor $r \in \{0.95, 0.98\}$ at each step.

4. RESULTS AND DISCUSSION

We recorded the 50 best layouts with the lowest fitness functions. We noticed that there is a common pattern that appeared several times among the best layouts. So, we considered this layout and compared it to the currently available Arabic Keyboards as shown in Table 1. It turned out that the optimized layout is 20% better than the currently available systems for mobile devices. It is even up to 33% better than the optimized Arabic Keyboards proposed in the literature, but this result is expected since that these keyboards were mainly designed for ten fingers use and then they attempt to distribute the typing effort among the two hands.

Arabic Keyboards	% improvement
Keyboard used in mobile devices	20%
Optimized Arabic Keyboard in [6]	33%
Ergonomic Arabic Keyboard in [7]	17%

Table 1: Comparison between Arabic keyboards.

Actually, there is no Arabic keyboard optimized for the applications using one pointer data entry in order to compare with. So we intend to consider other optimization frameworks based on metaheuristic like Genetic algorithms and compare them in solving the underlying QAP. The Genetic algorithm may be hybridized with simulated annealing to maximize its chances of obtaining good solutions.

Although our focus was on the minimization of the movement time, the designed layout will be also evaluated from a practical point of view using Eye Tracking technology. In fact, the proposed optimized layouts in the literature have been tested only from a mathematical point of view, and there is no study up-to-date about the end-users feedback and satisfaction. We propose to use the eye tracker device and observational studies in order to examine the user's experience to determine the most comfortable design.

5. REFERENCES

[1] A. Al-Wabil, A. Al-Issa, I. Hazzaa, M. Al-Humaimeedi, L. Al-Tamimi, and B. Al-Kadhi. Optimizing gaze typing for people with severe motor disabilities: the iwriter arabic interface. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility*, ASSETS '12, pages 261–262, New York, NY, USA, 2012. ACM.

[2] R. Burkard, M. Dell'Amico, and S. Martello. *Assignment Problems*. SIAM e-books. Society for Industrial and Applied Mathematics (SIAM), 2009.

[3] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381–391, 1954.

[4] Y. Li, L. Chen, and R. S. Goonetilleke. A heuristic based approach to optimize keyboard design for single-finger keying applications. *International Journal of Industrial Ergonomics*, 36(8):695 – 704, 2006.

[5] I. S. MacKenzie, A. Sellen, and W. Buxton. A comparison of input devices in elemental pointing and dragging tasks. *Proceedings of ACM CHI 91 Conference on Human Factors in Computing Systems*, pages 161–166, 1991.

[6] T. Malas, S. Taifour, and G. Abandah. Toward optimal arabic keyboard layout using genetic algorithm. In *Proceedings of the 9th Int Middle Eastern Multiconference on Simulation and Modeling (MESM)*, pages 50–54, Amman, Jordan, 2008.

[7] M. Osman. Ergonomic arabic keyboard. *http://iptl.ksu.edu.sa/sites/iptl.ksu.edu.sa/files/USD667414S1.pdf.*