Available online at www.sciencedirect.com

SCIENCE DIRECT®

International Journal of Industrial Ergonomics 36 (2006) 695–704

International Journal of INDUSTRIAL ERGONOMICS

www.elsevier.com/locate/ergon

A heuristic-based approach to optimize keyboard design for single-finger keying applications

Yanzhi Li, Lijuan Chen, Ravindra S. Goonetilleke*

Department of Industrial Engineering and Logistics Management, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong

Received 16 December 2005; received in revised form 20 April 2006; accepted 20 April 2006
Available online 15 June 2006

Abstract

With the popularity of mobile devices, designing a keyboard for users who can only operate with a pointer or the eyes is a challenging task. In this paper we present a methodology and keyboard designs that attempt to optimize the key arrangement so that movement time, as defined by Fitts' law, is minimized. Numerical simulations show that the present day keyboards and arrangements are not optimal for such applications. Three keyboard shapes, which included an existing design and two others were selected based on symmetry and compactness and were compared with two soft keyboards, FITALY and Metropolis. An integer programming model was formulated considering the character transition frequency of words in the English language. The problem was solved using a simulated annealing heuristic. One of the proposed designs called "YLAROF" with a rectangular design and a symmetric layout with characters laid in an "I" shape showed stable and consistent performance, when tested with 20 test cases.

Relevance to industry

When holding a mobile device, only one hand or one pointer is available for data input. Thus, it is important to minimize movement time in order to improve performance and minimize potential fatigue. The methodology presented and the proposed keyboard designs can be used and configured in existing products quite easily.

Keywords: Keyboard design; Fitts law; QWERTY; Dvorak; Gaze-control; Pointer-based control; FITALY; Metropolis

1. Introduction

The character arrangement on a keyboard can affect a person's comfort and typing performance (Wagner et al., 2003). The two most popular keyboards are the QWERTY and Dvorak keyboards. The QWERTY keyboard was introduced in 1872 and was primarily designed to slow down the typing speed so as to reduce jamming of mechanical parts. However, such a layout has been controversial in computer applications as it reduces typing speed as well as accuracy (Zecevic et al., 2000; Nevala-Puranen et al., 2003). The Dvorak keyboard developed in 1932, where the vowels are placed in the home row is claimed to overcome most of the limitations of the QWERTY keyboard (Brooks, 2005). However, the Dvorak keyboard has not completely replaced the QWERTY keyboard primarily due to long-term adaptation of the QWERTY keyboard. In addition, the level of enhancement with a Dvorak keyboard has not really justified the time and cost of retraining users (Sanders and McCormick, 1993).

Research related to keyboard arrangements has been conducted for differing reasons (Eggers et al., 2003). Anon (1999) has shown that good keyboard design can reduce the health risks and pain when involved with computer related work. Movement time, which is representative of speed of performance, has been predominantly used to evaluate keyboard designs. An excellent review is in MacKenzie and Soukoreff (2002). The FITALY (2006) keyboard has been designed based on the frequencies of the

*Corresponding author. Tel.: +852 23587109; fax: +852 23580062.
E-mail addresses: ieyanzhi@ust.hk (Y. Li), ljchen@ust.hk (L. Chen), ravindra@ust.hk (R.S. Goonetilleke).

0169-8141/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.ergon.2006.04.009

characters in the English language and has been based on the Brown Corpus for the English language. The most frequent characters are located in the middle of the keyboard. The lower frequency characters are placed towards the corners of the keyboard. In addition to mean tapping time (s) of characters, a transformation of which is words per minute (wpm) ($=60/5t$, assuming five characters per word) is also used for the comparison of different keyboards. Wagner et al. (2003) and Eggers et al. (2003) proposed a keyboard design methodology to minimize a function consisting of multiple criteria that included typing performance as well as ergonomic criteria. The ergonomic criteria were based on the load on fingers, number of key hits, hand alternation, avoidance of key hits by the same finger, avoiding large steps and a key hit direction from little finger to thumb. They designed an arrangement for the so-called ergonomic physical keyboard layout (ECP) and claimed that the design is better than the QWERTY keyboard by more than 50% and also slightly better than the Dvorak keyboard by about 1.9%. However, it can be seen that such a keyboard would not be appropriate for single finger use due to the criteria considered in the evaluation. Of course, there have been efforts to design keyboards for mobile phones, personal digital assistants (PDA) and the like so that they are easy to use by a novice user (Magnien et al., 2004). But increasing speed with a single finger hit has been a challenge even though many different types of keyboards are available (Sforh, 2005).

In this paper we provide a methodology to determine a keyboard arrangement for people using mobile devices such as PDA or for special groups of users such as the disabled or pilots who need to use gaze-controlled execution where actions are predominantly using just a single "pointer". Several different keyboards for single-finger keyboard (SFK) entry or for stylus-based text entry such as with the use of pen or stylus have been proposed in the literature. These include the ABC layout, FITALY, OPTI, Metropolis, Hooke, Lewis keyboards and many more (MacKenzie and Soukoreff, 2002). The manufacturer claims that the FITALY keyboard minimizes finger movement and also hand movement. The FITALY keyboard is 5×6 matrix for the alphabetical part of the keys. In addition, it is claimed that the layout of the keys is based on the frequencies of characters. The Hooke keyboard was based on minimizing the distance between likely characters using a greedy algorithm. The Metropolis keyboard (Zhai et al., 2000) on the other hand uses a random-walk strategy rather than a greedy algorithm. The Lewis keyboard uses an ad hoc method to minimize distances of the strongly associated pairs of characters. For such applications, a well-designed keyboard that minimizes pointer or finger or eye movement is critical to reduce fatigue and possibly reduce errors as well. Technology for gaze-controlled keyboard (GCK) is near maturity. Salcucci and Anderson (2000) designed an eye-controlled interface which uses gaze input effectively. Lankford (2000) demonstrated a system called "ERICA" which allows input using one's eyes in the Windows environment. Ward et al. (2002) showed that a two-dimensional eye-tracker can be used to reach an input speed of about 20 wpm for a novice user. All of such applications and demonstrations show the rapidly advancing techniques for numerous applications, especially suitable for disabled users. In a GCK the increased fixation duration, which can be programmed, is the signal to designate a "key-press". To minimize the repetitiveness of eye and head movement, which can give rise to fatigue and even injury, designing a keyboard that minimizes movement time can be very useful. Present day keyboards cannot be easily adopted for SFK or GCK gaze-controlled applications for many different reasons: (1) Due to space considerations, a SFK would not replicate a key unlike on a regular keyboard where keys such as "Control", "Shift", "Alt", etc. are repeated so that they can be used with alternate hands. (2) Only one key of an SFK can be "pressed" at any one time. (3) Symmetry and compactness constrain the shape of SFK, whereas a traditional keyboard has been designed with three rows to achieve efficiency using fingers of both hands.

Thus, we propose a methodology that can be used primarily to design and evaluate keyboard arrangements. Shieh and Lin (1999) have shown that the keytapping times are dependent on the hand movement and the characteristics of the material that is typed. This gives a unique way to formulate a performance function if the character transition frequencies (CTFs) can be determined (MacKenzie and Soukoreff, 2002). Our approach is based on this fundamental concept.

2. Methodology

The most important step of this methodology is establishing a meaningful objective function.

2.1. Movement time

Movement time can be estimated in many different ways. Wagner et al. (2003) and Eggers et al. (2003) used rectilinear distance between two keys as a measurement, with the assumption that movement time is proportional to the movement distance. Rosen et al. (1986) proposed the following empirical model:

$$T = a + bA + c/W,$$

where T is the movement time, a, b, c are constants in a particular context, A is the distance between two targets and W is the width of the targets.

Fitts (first appeared in Fitts, 1954 and reprinted in Fitts, 1992) proposed that the time to move between a starting point and a target (T) is proportional to the index of difficulty (I) where $I = \log_2(2A/W)$

$$T = a + b\log_2[2A/W] \quad \text{where } a \text{ and } b \text{ are constants.}$$

Subsequently, two variations to the above formulation have been proposed where $I = \log_2(A/W + c)$ that provides a better fit to data and one that always gives a positive rating for the index of difficulty. The case with $c = 0.5$ is known as the Welford (1960) formulation while $c = 1$ is known as the Shannon formulation. The Shannon formulation given below is supposedly superior to the other two forms as it provides a better fit with observations (MacKenzie, 1989):

$$T = a + b \log_2[A/W + 1] \tag{1}$$

where a and b are constants.

Except for the space bar, we will assume that all keys are square and of dimension 1×1 units. In other words, assume $W = 1$. The space bar in keyboards I, II and III is wider than the other keys. Hence the distance between a character key and the space key varies depending on the point at which the space key is tapped. The approximation used in MacKenzie and Zhang (1999) wherein the space key is divided into multiple segments each of width equal to a character key (for example, the space key in keyboard I is divided into six segments with each being a 1×1 square) was used. The distance between a character key and the space key is then calculated as the minimum among the distances between the character key and the divided space segments.

Owing to individual variations, the intercept, a may not be exactly zero, but ought to be relatively small and a large value for the intercept is an indication of some problem with the experimental methodology (MacKenzie, 1989). If the intercept i not equal to zero, then $T > 0$ even with $I = 0$. Hence it is reasonable to use the form $T = b \log_2[A/W + 1]$ proposed by Card et al. (1983). The constant b having units of time/bits has been found to be relatively constant for the same type of task (examples include Fitts and Peterson, 1964; Welford, 1968, etc.). More recently, MacKenzie et al. (1991) found b to be equal to 1/4.9 bps for pointing tasks using a stylus as a computer input device. Due to this relatively constant value for stylus related tasks, it is appropriate to evaluate and make conclusions based on $T_{ij} = b \log_2[A_{ij}/W + 1]$, where i,j are two keys, A_{ij} is the distance between the centers of the keys and W is the width of a key (the exact definition of W is specified in Section 2.3).

In order to make comparisons with previous literature such as MacKenzie and Zhang (1999) and Zhai et al. (2000), mean time (s) for typing one character $t_{ij} = 1/4.9[\log_2(A_{ij}/W + 1)]$ was used. In addition, the wpm measure was also used for comparison purposes where five characters were assumed to be equal to one word. In other words, wpm $= 60/5t$. In the special case where a character is repeated or pressed twice (that is, $i = j$), the above equation gives $T = 0$. However, in this case a in Fitts' equation was set to 0.127 s (that is, $t_{ii} = 0.127$ s) in order to compare the results with MacKenzie and Zhang (1999) and Zhai et al. (2000).

2.2. Character transition frequency

CTF is an important component for the design of a keyboard (Shieh and Lin, 1999). Characters with higher frequency transitions from one character to another should be positioned close together to minimize movement. In the ideal case, the best keyboard is one that is specific for the material been input. However, such a solution will not have universal appeal. Thus, we used the 15,000 commonest words from the British National Corpus (known as the BNC), which has a count of 100 million English words—equivalent to the full text of about a thousand books in order to obtain the necessary information. The statistics on word frequency is available at the Audience Dialogue website (Audience Dialogue, 2005) based on which it is easy to calculate CTF, which is the transition frequency from one character to another. Note that the transition among the words need not be considered as a space is intertwined between words. However, we do need to consider the transition between characters and the "space" key. We make the assumption that there is a space before and after each word, which is reasonable except that we consider punctuation at the end of a sentence as a "space" as well.

2.3. Keyboard shapes

Keyboards such as QWERTY and Dvorak have three rows with a home row of nine letters and two other rows to make up the 26 characters. However, such a layout may not be suitable for a SFK or GCK where the keys are pressed using a single "pointer". Three keyboard shapes are proposed and evaluated as shown in Fig. 1. Note that the keys are 1×1 unit squares.

Shape I is the present day keyboard (the 1/3 key indent between the first and second rows and the 1/2 key indent between the second and third rows were considered in the distance calculations). Shapes II and III are symmetric and compact so that they are suitable for mobile devices.

Shape II was based on a patented SFK called FITALY (US Patent Number 5,487,616) by Textware Solutions (FITALY, 2006; MacKenzie et al., 1999; Zhai et al., 2002). The keyboard is primarily designed for a probe or a computer with a touch screen and its layout is shown in Fig. 2.

The manufacturer claims that the FITALY keyboard minimizes finger movement and also hand movement. The FITALY keyboard is 5×6 matrix for the alphabetical part of the keys. In addition, it is claimed that the layout of the keys is based on the frequencies of characters. The FITALY and OPTI keyboards are ones where the characters are laid out in a 5×6 matrix. Thus, we adopted the 5×6 configuration as keyboard II. On the other hand, the Metropolis keyboard (Fig. 2) is one with keys in a 5×7 configuration. Unlike the manually designed keyboards, the Metropolis and Hooke keyboards do not have the characters in a rectangular format (Zhai et al., 2000). Thus,

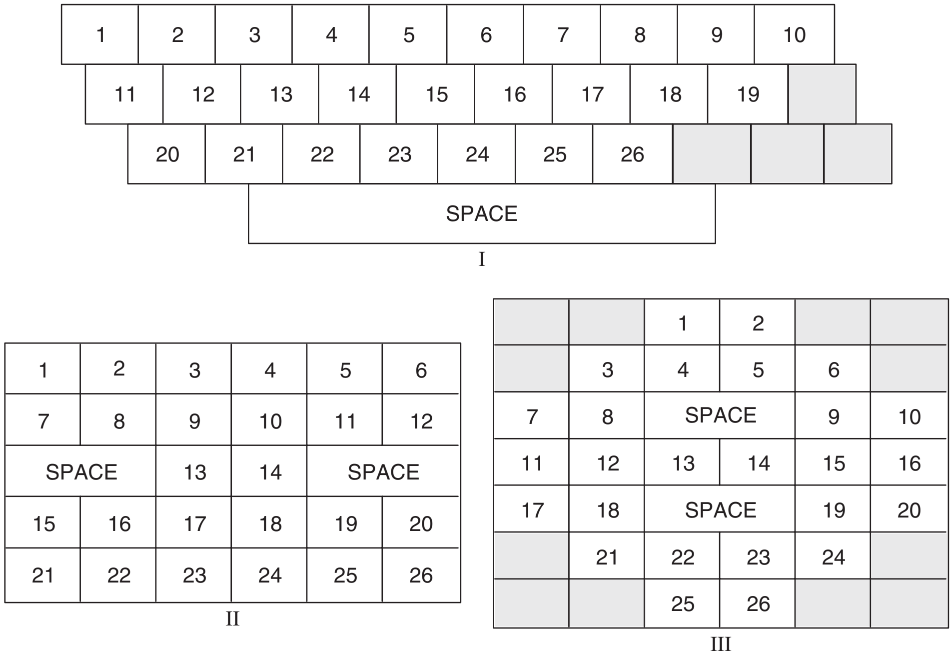


Fig. 1. Keyboard shapes I, II and III with keys numbered prior to character assignment.

a relatively larger grid of 7×6 was also investigated and is denoted as keyboard III.

The keys were numbered for the assignment process (Fig. 1). The layout (which denotes the physical shape of a keyboard as opposed to "arrangement", which denotes the assignment of characters to keys) of the Dvorak and QWERTY keyboards are similar except for the arrangement of keys, which is different. In order to apply Fitts' law, the height and width of all character keys were set to be 1. Hence W was set to be 1 in the calculation of movement time as it is supposed to be the smaller value of target width or height, as suggested by MacKenzie and Buxton (1994).

3. Solution approach

An integer programming (IP) model was formulated and due to the scale of the IP model, a heuristic approach was used to solve it.

3.1. IP model

Suppose we only consider 26 letters and the other keys (special characters and punctuation marks) were assigned manually, we propose the following formulation.

Let the subscripts i, j denote letters and m, n denote keys. Given a keyboard layout, the traveling cost between any two keys is fixed, which can be calculated based on Fitts' law or simply the distance between centers of the two keys. Assume the movement time (in "bits") between any two keys m and n is given as c_{mn} (since we eliminated a and b and use equation $T = \log_2[A/W + 1]$ instead, as the optimization will be independent of these constants) and transition frequency from letter i to j is q_{ij}. Note that both a and b were used in the wpm calculations as indicated before.

Decision variables:

I_{im}: 1 if letter i is assigned to key m; 0 otherwise.

J_{ijmn}: 1 if letters i and j are assigned to keys m and n, respectively; 0 otherwise.

$$f = \min \sum_{i \neq j} \sum_{m \neq n} c_{mn} q_{ij} J_{ijmn} \tag{2}$$

s.t.

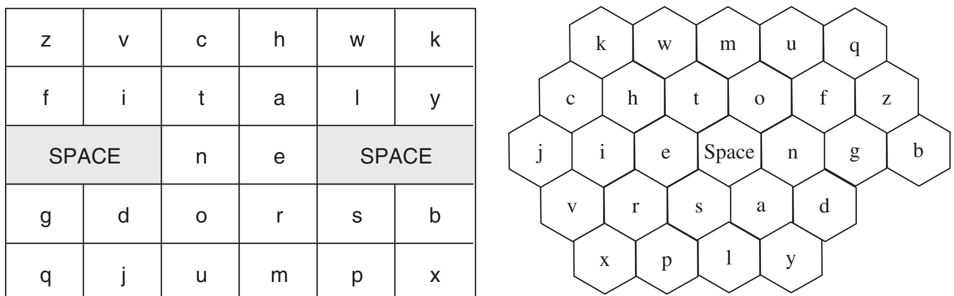
$$\sum_i I_{im} = 1, \quad \forall m, \tag{3}$$

$$\sum_m I_{im} = 1, \quad \forall i, \tag{4}$$

$$I_{im} + I_{jn} \leqslant 1 + J_{ijmn}, \quad \forall i, j, m, n, \tag{5}$$

$$I_{im}, J_{ijmn} \in \{0, 1\}, \quad \forall i, j, m, n. \tag{6}$$

The above IP model has approximately $26^4 + 26^2 + 26^2 = 458\,328$ binary variables and thus not easy to solve. We attempted to solve the above using CPLEX 9.0 on a PC with 2.5 GHz CPU and 1 GB RAM. Even though the program ran for over 60 h and the solution improved each time, it was unable to attain a solution within reasonable convergence criteria. Hence a heuristic approach was used to solve the above.



FITALY Metropolis

Fig. 2. The FITALY and Metropolis keyboard layout.

3.2. *A two-stage heuristic*

A simulated annealing (SA) (Kirkpatrick et al., 1983) two-stage heuristic was used to solve the problem. SA is an analog of the physical annealing procedure. It is a metaheuristic to help a local search procedure escape from local optimum. It enables the search procedure to explore the whole solution space by introducing a probability of allowing poor moves so that a solution with minimum energy may be found.

The process starts with a random initial solution. After numbering the keys as shown in Fig. 1, a solution can be represented as an array describing the assigned character of each position (see objective function given in Eq. (2)). A two-stage process is then applied.

Stage 1. Exact search. In this stage the neighborhood of two-exchanges is considered, i.e., a new solution is obtained by exchanging the position of any two characters. Thus, the neighborhood size of a solution is $26 \times 25/2 = 325$. A greedy search is performed for a better solution by enumerating until an improved solution cannot be obtained in 10 consecutive attempts. This is an important step and has not been used in the development of the Metropolis keyboard (Zhai et al., 2000).

Stage 2. Simulated annealing stage. A reheating scheme (Anagnostopoulos et al., 2006) is used during the second stage to further increase the chance of finding a global optimum. At first, the temperature (t) is set to a high value so that many "poor" keyboard arrangements that have higher objective functions are accepted. This allows a large part of the solution space to be assessed. The temperature, t, is then gradually decreased at each step to "$t \times r$" according to an annealing schedule where r is ratio of temperature decrease. At each decrease, we accept keyboard arrangements that have a lower objective function. We reheated five times (i.e., $M = 5$). The parameters including t, L, M and r are central to the performance of the algorithm. After extensive experiments, the initial temperature t was set to 1×10^7, number of steps (L) at the same temperature was set to 5, number of reheating schemes $M = 5$ and $r = 0.95$ (refer to Fig. 3 for detailed procedure, where $f(S)$ is the objection function value for a solution S calculated using Eq. (2)).

4. Numerical experiments

4.1. *Results*

The algorithm attains a solution in less than 1 min. The keyboard arrangements for the three shapes, i.e., I, II and III, are shown in Fig. 4 and are hereby called "RANI", "YLAROF", and "HERF" based on the character positioning in the one before last row. The following objective function was used to evaluate the keyboard shapes and layouts:

$$\sum_{i,j} c_{\pi_i \pi_j} q_{ij}, \tag{7}$$

where i and j are any two letters, π_i, π_j are the keys assigned to them, $c_{\pi_i \pi_j}$ is equal to $\log_2[A_{ij}/W + 1]$ and q_{ij} is the transition frequency between i and j obtained for BNC database. The above function accounts for the movement time associated with the transitions between words using a large database of English words found in BNC.

The objective function values from Eq. (7) (and the wpm calculated with a and b) for the proposed and other keyboards using the BNC database are as follows: keyboard I is 6.21874E + 8 bits (40.1127 wpm); keyboard II is 5.73833E + 08 bits (43.4451 wpm); keyboard III is 5.92030E + 08 bits (42.1196); QWERTY is 8.02146E + 08 bits (31.1476 wpm); Dvorak is 8.73689E + 08 bits (28.61 wpm); FITALY is 5.85167E + 08 bits (42.6099 wpm); and Metropolis is 5.82754E + 08 bits (42.785 wpm). Accordingly, keyboard II has the lowest movement time with the Metropolis and FITALY keyboards being second and third, respectively. All three new designs (I = "RANI"; II = "YLAROF"; III = "HERF") outperform the QWERTY and Dvorak keyboards. Interestingly, for single-finger input, QWERTY appears better than the Dvorak keyboard, which is not very surprising as the advantage of the vowels in the home row is lost with the use of a single-finger key-press.

4.2. *Validation*

Even though the configurations obtained for each of the three shapes are based on BNC data, it is still necessary to

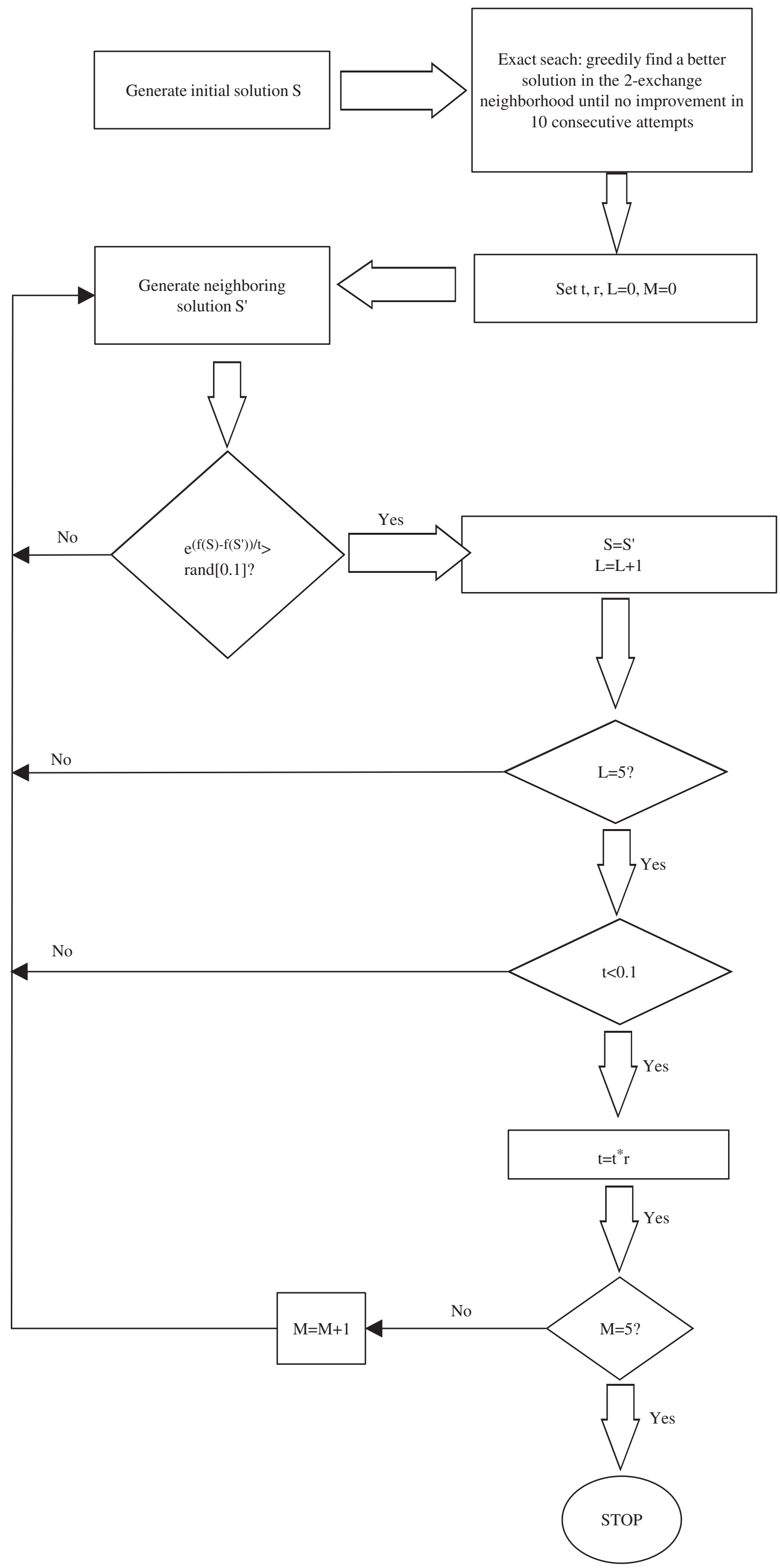


Fig. 3. Algorithm flow chart.

validate the arrangement using test cases. For this purpose, we digested 20 essays from BBC.com and Time.com in different disciplines. A sample test case is given in the Appendix. Note that the numbers and punctuations are ignored and all the letters are transformed to lower case to obtain the character transitions. The results are shown in Table 1 and Fig. 5, with each number representing the wpm for typing the given text, using wpm $= 60/5t$ where

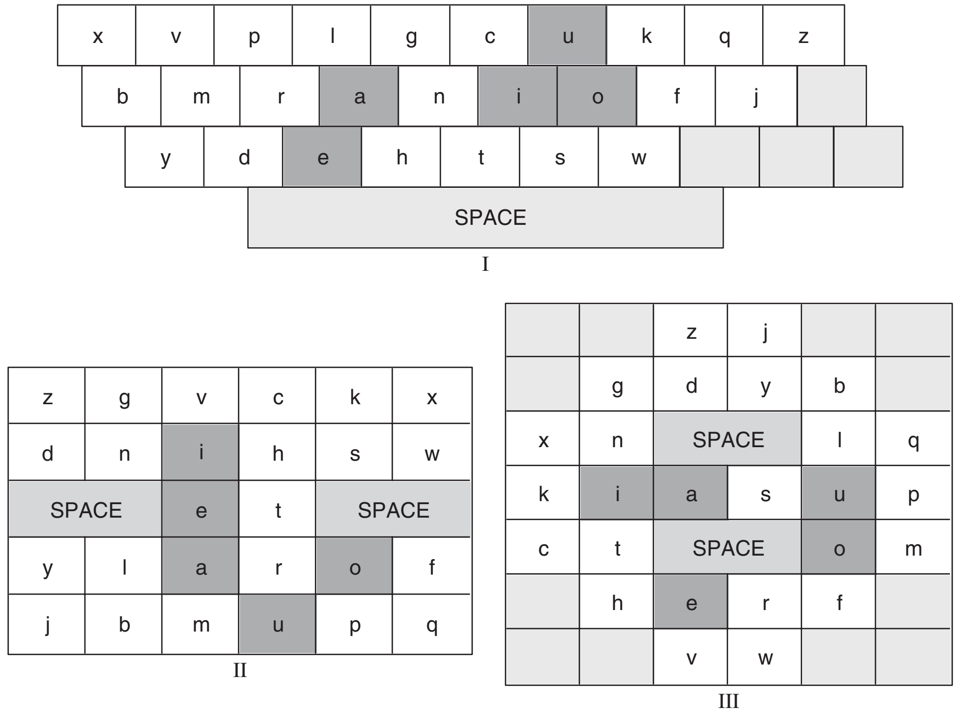


Fig. 4. Optimal arrangements for the keyboard shapes I, II and III.

$t_{ij} = 1/4.9[\log_2(A_{ij}/W + 1)]\ \forall i \neq j$ and $t_{ii} = 0.127\,\mathrm{s}$. It is evident that keyboard II (YLAROF) is superior to all others for all the 20 test cases followed by the Metropolis, FITALY, III (HERF), I (RANI), QWERTY and Dvorak keyboards as before. A 2-way (keyboard × text type) ANOVA on wpm showed significant differences for keyboard ($F(6, 91) = 6177.12$; $p < 0.0001$) and text type ($F(6, 91) = 6.39$; $p < 0.0001$) but no significant keyboard × text type interactions. Each keyboard was significantly different from each other with the highest to lowest wpm order of YLAROF, Metropolis, FITALY, HERF, RANI, QWERTY and Dvorak, respectively. The performance stability of keyboard II (YLAROF) for the differing test cases is somewhat surprising as the passages are from different disciplines and are of varying length. The stability and the improved performance of keyboard II (YLAROF) clearly brings out the need to consider specialized keyboards for single pointer entry and gaze-controlled applications.

5. Discussion and conclusions

In this study a keyboard arrangement, that is superior to those that have been proposed in the literature, has been determined for single-finger or gaze-controlled applications based on movement time. Since the proposed integer programming model was difficult to solve, a two-stage heuristic was used which gave a solution in a relatively short time with exact search in the first stage and a simulated annealing-based search in the second stage. Extensive tests show that keyboard II has consistent performance with varying types of manuscripts and clearly outperforms the QWERTY and Dvorak keyboards primarily because the existing keyboards rely on multiple finger use of both hands.

The approach used in the model is based on Fitts' law and character transition frequencies of a very large database. To date, we are unaware of such an approach in order to design a keyboard for any language. The arrangements of keys in all three shapes show a familiar pattern. The five vowels are in close proximity even though they are not exactly next to each other, which brings out an important aspect that even though they are vowels that have relatively high frequency, they do not have similar character transition frequencies. This is possibly an aspect that may have been overlooked in the design of a Dvorak keyboard.

The Metropolis keyboard is a close second primarily due to the hexagonal shape of the keys. Due to this feature, we trained the Metropolis keyboard on the BNC database and was able to improve on the key arrangement. The improved Metropolis keyboard is shown in Fig. 6 and the wpm values for the improved keyboard are shown in Table 1. This shows that the keyboard layout plays an important role in the optimum key arrangement. The proposed method determines the optimum key arrangement for a given layout (for example, *RANI* for layout I, *YLAROF* for layout II, and *HERF* for layout III) and is not guaranteed to be a truly optimal configuration for every possible layout. It is difficult to imagine attaining a wpm value of around 40 for the proposed keyboards especially with single-finger keying. However, this value is

Table 1
WPM values for the 20 test cases

Test cases	Keyboard I	Keyboard II	Keyboard III	QWERTY keyboard	Dvorak keyboard	FITALY keyboard	Metropolis keyboard	Trained Metropolis	Text type
1	39.7238	42.7861	40.9114	30.7944	27.8714	41.6529	42.1894	42.8312	History
2	40.1036	43.1995	42.0969	31.8431	28.0271	42.169	43.1377	43.1547	History
3	40.2732	43.1746	41.9571	30.8423	28.575	41.6263	43.0394	43.6302	History
4	39.9715	42.8342	41.3341	30.9877	27.9892	42.0139	42.4528	43.2041	Business
5	39.4812	42.6081	40.9543	30.7736	28.1008	41.5435	42.0952	42.8236	Business
6	39.7651	43.2191	41.4336	30.7545	28.2483	42.2013	42.7561	43.1285	Business
7	38.593	42.1089	40.8861	31.1309	28.3692	41.7875	41.8345	42.5953	Business
8	39.9194	43.1172	41.845	30.927	28.0373	41.9324	42.758	43.6127	Art
9	39.4984	42.6163	41.3745	31.0706	28.1142	42.3735	42.4167	43.04	Art
10	39.6673	42.791	41.2882	31.0466	28.0514	42.191	41.789	42.8158	Art
11	39.6878	43.3303	41.7554	31.3699	28.5559	42.1466	42.3948	43.7384	Art
12	39.882	43.2164	41.8737	31.1593	28.6405	42.8212	43.0676	43.8841	News
13	39.5704	43.3212	42.1988	31.0477	28.4211	42.2439	42.3225	43.426	News
14	39.3632	42.9813	41.6941	31.2947	28.5495	41.9825	42.3742	43.4287	Politics
15	39.9826	43.1182	41.6338	31.5004	28.7732	42.4128	42.7627	43.2965	Politics
16	39.5796	42.9898	41.2998	30.962	28.1634	42.1178	42.3436	43.0532	Religion
17	39.4548	42.6014	40.9371	30.9782	28.0289	41.6101	42.1535	42.7246	Religion
18	39.2402	42.6578	41.0517	30.8379	28.2231	41.7685	42.3763	42.5965	Religion
19	38.892	42.7136	41.2097	30.951	28.4228	41.6138	42.0143	42.9688	World
20	40.0214	43.0817	41.5527	30.72	27.7819	42.2179	42.2907	43.2631	World

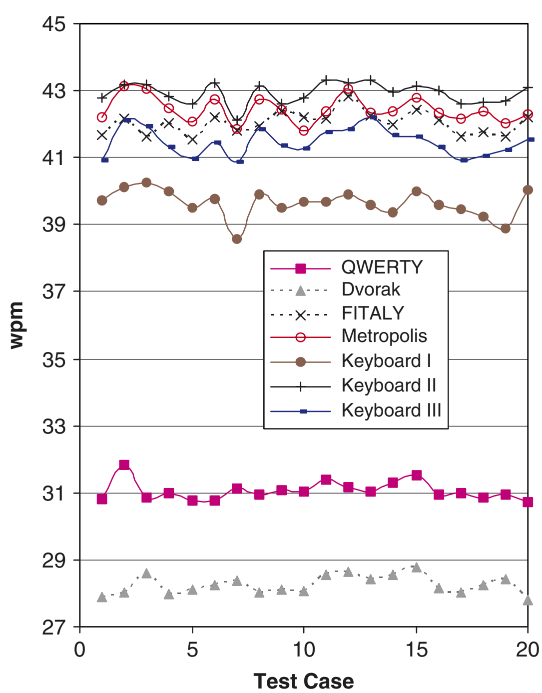


Fig. 5. The predicted wpm of each keyboard for differing test cases.

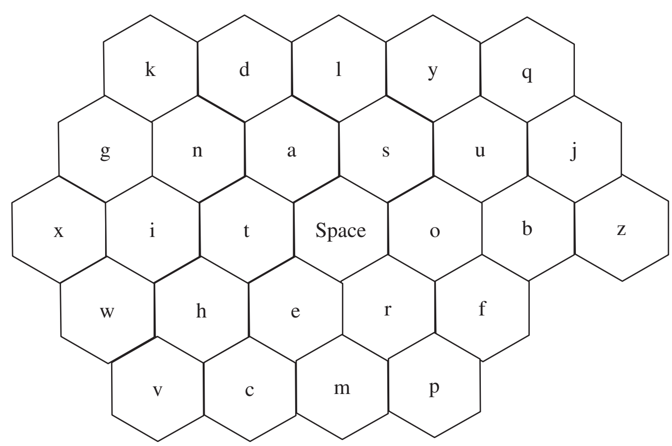


Fig. 6. Improved Metropolis keyboard.

dependent on the value of b used in Fitts' equation which was set to be 1/4.9 in order to compare with previous research. Since b is a constant, it will not affect the methodology but will change the wpm values if changed to a different value.

The proposed methodology does have its shortcomings. If fingers are used for text entry rather than a stylus, sometimes, both thumbs tend to be used when holding the device in one hand. This type of entry may need a different formulation and is thus a limitation of our study. In addition, we have primarily considered the alphabetical characters and ignored numbers and punctuation marks as it is difficult to estimate their frequencies. However, they may be incorporated into a few keys such as in a mobile phone. Also, only three shapes were evaluated even though shapes II and III can be easily adapted in rectangular designs. There may be a need to look at other shapes as well depending on available physical shapes. Even though the movement time is optimized, users may have a learning curve with achieving optimal performance due to unfamiliarity of the keys. Knowing the transitions, it is possible to assist the user to configure the remaining characters with the aid of the first few characters. It may also be necessary to test prototype keyboards to assess human performance over long periods of time in order to make conclusive judgments about the effectiveness of different keyboards. The methodology has focused on optimizing movement time assuming the validity of a variation of Fitts' law. Many who use alternative keyboards (especially gaze based) may have physical disabilities or limitations and these may have an affect on the validity of Fitts' law and may require other considerations as well. Nevertheless, the proposed approach can be a means to assess the theoretical foundations of the design of a keyboard for improved speed.

Appendix. Example test case

The test cases can be found in Li (2005). Test case 1 is shown below and is digested from BBC (2005).

The life of Gnaeus Julius Agricola is known to us today because his son-in-law, the famous Roman historian, Tacitus, left a detailed biography of the Roman general. Agricola is recorded in British history because he conquered parts of Wales and Scotland but he won other impressive titles including Quaestor in Asia (AD 64), People's Tribune (AD 66) and Praetor (AD 68). It was during the civil war of AD 69 that Agricola supported Vespasian, who in turn appointed him commander of a force headed for the British Isles. Eight years later, Agricola was made Governor of Britain. In between he returned to Rome in AD 73 and served as Governor of Aquitania for three years. Back in Britain, Agricola set about conquering more remote regions in northern England, Scotland and Wales. According to Tacitus he crossed the Menai Straits to take Anglesey, reportedly massacring the island's inhabitants who were of the Druid faith. Tacitus does not spare us the details, giving a vivid account of wild-haired women and barbarian Druids who created a formidable line on the shore opposite the mainland. The Druids were nonetheless conquered and their base on Mona (modern-day Anglesey) was broken up. From AD 79–80, Agricola moved north to Scotland where he consolidated Roman military control of the Forth-Clyde line. He was the one who masterminded the building of a string of forts across the country from west to east. From AD 81–83, Agricola campaigned north of the Forth-Clyde line and confronted the Caledonii (under Calgacus) at the battle of Mons Graupius. Although the result was indecisive, Agricola's effort paved the way for the creation of the most northerly legionary fortress of the Roman Empire at Inchtuthill in Perthshire. Recalled to Rome, Agricola lived in retirement, having refused the proconsulship of Asia. He died in AD 93.

References

Anagnostopoulos, A., Michel, L., Hentenryck, P.V., Vergados, Y., 2006. A simulated annealing approach to the traveling tournament problem. Journal of Scheduling 9 (2), 177–193.

Anon, 1999. Keyboard design is effective in decreasing user pain. IIE Solutions 31 (7), 14.

Audience Dialogue, 2005. ⟨http://www.audiencedialogue.org/susteng.html⟩, December 15, 2005.

BBC, 2005. ⟨http://www.bbc.co.uk/history/historic_figures/agricola_gnaeus_julius.shtml⟩, December 15, 2005.

Brooks, M., 2005. ⟨http://www.mwbrooks.com/dvorak/⟩, December 15, 2005.

Card, S.K., Moran, T.P., Newell, A., 1983. The Psychology of Human–Computer Interaction. Lawrence Erlbaum, Hillsdale, NJ.

Eggers, J., Feillet, D., Kehl, S., Wagner, M., Yannou, B., 2003. Optimization of the keyboard arrangement problem using an ant colony algorithm. European Journal of Operational Research 148 (3), 672–686.

FITALY, 2006. One finger keyboard. ⟨http://www.fitaly.com/fitaly/fitaly.htm⟩, accessed on April 2, 2006.

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

Fitts, P.M., 1992. The information capacity of the human motor system in controlling the amplitude of movement. Journal of Experimental Psychology: General 121 (3), 262–269.

Fitts, P.M., Peterson, J.R., 1964. Information capacity of discrete motor responses. Journal of Experimental Psychology 67, 103–112.

Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220 (4598), 671–680.

Lankford, C., 2000. Effective eye-gaze input into windows™. Eye-tracking Research and Application Symposium, Sponsored by ACM SIGCHI and SIGGRAPH, West Palm Beach, Florida.

Li, Y., 2005. ⟨http://ihome.ust.hk/~ieyanzhi/kap/testcases.zip⟩, December 15, 2005.

MacKenzie, I., 1989. A note on the information-theoretic basis for Fitts' law. Journal of Motor Behavior 21, 323–330.

MacKenzie, I.S., Buxton, W., 1994. Prediction of pointing and dragging times in graphical user interfaces. Interacting with Computers 6 (2), 213–227.

MacKenzie, I.S., Soukoreff, R.W., 2002. Text entry for mobile computing: models and methods, theory and practice. Human–Computer Interaction 17, 147–198.

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

MacKenzie, I.S., Sellen, A., Buxton, W., 1991. A comparison of input devices in elemental pointing and dragging tasks. Proceedings of the CHI '91: ACM Conference on Human Factors in Computing Systems, pp. 161–166.

MacKenzie, I.S., Zhang, S., Soukoreff, R.W., 1999. Text entry using soft keyboards. Behaviour and Information Technology 18 (4), 235–244.

Magnien, L., Bouraoui, J., Vigouroux, N., 2004. Mobile text input with soft keyboards: optimization by means of visual clues. Lecture Notes in Computer Science, vol. 3160. Springer, Berlin, pp. 337–341.

Nevala-Puranen, N., Pakarinen, K., Louhevaara, V., 2003. Ergonomic intervention on neck, shoulder and arm symptoms of newspaper employees in work with visual display units. International Journal of Industrial Ergonomics 31 (1), 1–10.

Rosen, M., Goodenough-Trepagnier, C., Getschow, C., Felts, T., 1986. Qualitative aspects of Fitt's task performance by motor-disabled individuals. Ninth Annual Conference on Rehabilitation Engineering, pp. 161–163.

Salcucci, D.D., Anderson, J.R., 2000. Intelligent gaze-added interface. Proceedings of CHI 2000 Conference on Human Factors in Computer Systems, ACM, New York.

Sanders, M.S., McCormick, E.J., 1993. Human Factors in Engineering Design. McGraw-Hill, New York.

Sforh, 2005. ⟨http://www.sforh.com/keyboard-idx.html⟩, December 15, 2005.

Shieh, K., Lin, C., 1999. A quantitative model for designing keyboard layout. Perceptual and Motor Skills 88 (1), 113–125.

Wagner, M., Yannou, B., Kehl, S., Feillet, D., Eggers, J., 2003. Ergonomic modelling and optimization of the keyboard arrangement with an ant colony algorithm. Journal of Engineering Design 14 (2), 187–208.

Ward, D.J., Blackwell, A.F., MacKay, D.J.C., 2002. Dasher: a gesture-driven data entry interface for mobile computing. Human–Computer Interaction 17, 199–228.

Welford, A.T., 1960. The measurement of sensory-motor performance: survey and reappraisal of twelve years' progress. Ergonomics 3, 189–230.

Welford, A.T., 1968. Fundamentals of Skill. Methuen, London.

Zecevic, A., Miller, D., Harburn, K., 2000. An evaluation of the ergonomics of three computer keyboards. Ergonomics 43 (1), 55–72.

Zhai, S., Hunter, M., Smith, B.A., 2000. The Metropolis keyboard—an exploration of quantitative techniques for virtual keyboard design. Proceedings of the ACM Symposium on User Interface Software and Technology—UIST 2000, ACM, New York, pp. 119–128.

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