Revisiting and Validating a Model of Two–Thumb Text Entry

Edward Clarkson, Kent Lyons, James Clawson and Thad Starner
College of Computing and GVU Center, Georgia Institute of Technology
801 Atlantic Drive, Atlanta GA 30332-0280
{edcclark, kent, jamer, thad}@cc.gatech.edu

ABSTRACT

MacKenzie and Soukoreff have previously introduced a Fitts' Law–based performance model of expert two–thumb text entry on mini–QWERTY keyboards [4]. In this work we validate the original model and update it to account for observed behavior. We conclude by corroborating our updated version of the model with our empirical data. The result is a validated model of two-thumb text entry that can inform the design of mobile computing devices.

Author Keywords

Mini–QWERTY, Mobile Text Entry, Fitts' Law

ACM Classification Keywords

H.5.2: User Interfaces, Input devices and strategies

INTRODUCTION

Mini-QWERTY keyboards are miniature versions of the traditional desktop QWERTY keyboard and are used in many mobile devices. MacKenzie and Soukoreff introduced a Fitts' Law–based performance model of two–thumb text entry on mini–QWERTY keyboards [4]. Such a model can help inform the design of new mini-QWERTY keyboards by providing *a priori* predictions of expert performance. We have previously reported the results of a longitudinal study of typing rates on these kinds of keyboards [1]. In this paper we examine the original model in light of that study, suggest an alteration to improve its accuracy, and corroborate the updated model against our empirical data. The result is a validated model of expert two–thumb text entry, which may be a useful tool for mobile device designers.

Fitts' Law [2] is a performance model for aimed movement when users are experts at a given pointing task and cognitive overhead is not relevant. Silfverberg *et al.* used Fitts' Law to develop a model for expert text entry on a standard 12-key mobile phone keypad [6]. MacKenzie and Soukoreff proposed a similar model for two-thumb text entry on miniature keyboards [4]. They provide a general description of user/keyboard interaction and predicted an expert rate of 60.74 words per minute (wpm) for their example keyboard. However, they provide no empirical validation of this figure.

Figure 1. The mini–QWERTY keyboards used in our previous studies: Targus (left) and Dell (right).

In our previous experiment [1], participants used either a Dell or Targus brand mini-QWERTY keyboard for twenty 20-minute typing sessions to obtain empirical typing data. The Mackenzie and Soukoreff model predicts an expert rate of 57.88 wpm with the specifications of our Dell keyboard. At first glance, this seems to align relatively well with our study results: the mean typing rate after 400 minutes of practice for the Dell participants was 59.32 wpm, a difference from the model of less than 3%. However, a closer look at the results reveals discrepancies between the predicted and observed *inter-key times*—the time span between consecutive key presses.

TWO-THUMB TYPING MODEL OVERVIEW

Before discussing the details of the inter-key timing differences, the reader my find a brief review of the the MacKenzie and Soukoreff model helpful. The model takes as inputs a representation of a two–thumb keyboard layout, a word corpus and a key–thumb assignment table and produces a value representing the typing rate for an average expert user. The representation of the keyboard encodes the physical layout of the keys on the device. The word corpus can be any representative body of sentences, such as MacKenzie and Soukoreff's own corpus [5].

The model employs two fundamental times. The first time calculation represents the time for a thumb to move from one key to the next and is derived from Fitts' Law, using Fitts' Law coefficients from related work [6]. These transition times are denoted t_{fitts}. The second time employed is the minimum time between keystrokes using opposite thumbs (88 ms, denoted t_{min}). This time is half of the key repeat rate (176 ms) and is the minimum time it takes to press the same key twice in a row. The model uses these times together to calculate the time to press a sequence of keys.

The model begins by assigning a time to the first letter in a word. This time depends on which side of the keyboard the first letter is on and which thumb pressed the preceding space key. Using the British National Corpus, MacKenzie and Soukoreff calculated that the right thumb is used to press the space key 70.49% of the time. If the first key of a word is assigned to the right thumb, the time it takes to move from the space key to the first letter of the word will be the Fitts–modeled transition. If the left thumb pressed the space key, the right thumb can move in parallel but still takes at least t_{min}. When the first character is assigned to the left thumb, the weighting factors are simply reversed. T_n represents the predicted time for a word up to the n^{th} character.

$$T_1 \approx \begin{cases} 0.70 \cdot t_{fitts} + 0.30 \cdot t_{min} & \textit{if right thumb first} \\ 0.30 \cdot t_{fitts} + 0.70 \cdot t_{min} & \textit{if left thumb first} \end{cases}$$

For subsequent characters, if the current and previous keys are assigned to the same thumb, the prediction is increased by the time to move the thumb, $t_{fitts}(key_{n-1}, key_n)$. When the current and previous keys are assigned to opposite thumbs, the current key may have been located in the time it took for the opposite thumb to press the previous letter. The time for the opposite thumb to move from its former location (*k* keys ago) to the current key is $t_{fitts}(key_{n-k}, key_n)$. However, even if the thumb has reached the target, the time to activate the key is at least t_{min}:

$$T_n = \begin{cases} T_{n-1} + t_{fitts}(key_{n-1}, key_n) & \textit{same} \\ max \begin{pmatrix} T_{n-1} + t_{min}, \\ T_{n-k} + t_{fitts}(key_{n-k}, key_n) \end{pmatrix} & \textit{opposite} \end{cases} \quad (1)$$

INTER-KEY TIMING DISCREPANCIES

The model explicitly handles transitions between letters assigned to the same thumb separately from those assigned to different thumbs. As such, it is natural to assign inter-key times into categories based on the thumb assigned to the source and destination keys. There are four such categories: Left→Left; Right→Right; Left→Right; Right→Left. Note that the former two are modeled solely by Fitts' Law while the latter two are slightly more complex (see Eq. 1).

Thus, it may be useful to compare the differences between predicted and actual inter-key times for each of these four classes of transitions. Since the model is of expert use, we limit our data source to error-free sentences from Dell users' sessions 16-20 collected from our previous work [1]. Table 1 shows the average deviations of the observed inter-key times from the predicted values for each transition class (e.g., L→R). To calculate the table entries we first find the mean observed inter-key time and the model-predicted inter-key time for every key pair. We then subtract the predicted from the observed value for each key pair and average the deviations within each transition class. The figures in Table 1 are not weighted by frequency, but such an adjustment yields similar results.

Source \ Dest.	Left	Right
Left	(24)	73
Right	126	(38)

Table 1. Average deviation (in ms) of empirical inter-key times from predicted for each thumb transition type. Parenthesized values are slower than observed.

Table 1 shows the model consistently makes predictions which are *slower* than observed for Fitts' Law-modeled transitions (L→L and R→R) and *faster* than observed for cross-thumb transitions. Consequently, the model's relatively accurate prediction masks inaccuracies at a lower level; as such it is worthwhile to investigate possible improvements. One possibility is revisiting the assumption that each key is statically assigned to a particular thumb.

VARIABLE THUMB-KEY ASSIGNMENTS

The original model assumes static thumb–key assignments: the model designates the left thumb to press keys on the left half of the keyboard and the right thumb to press keys on the right half. While we did not collect direct evidence of actual thumb usage patterns, informal observation and anecdotal user comments suggest that keys in the middle of the keyboard may be pressed by either thumb. We refer to these keys and this kind of usage as *dynamic*, *variable* or *flexible*. Alternating thumb use allows users to perform thumb movements in parallel, which can increase typing rates in certain circumstances. Given our knowledge of this practice by users, we extended the model to incorporate the concept of keys that can be operated by either thumb.

The basic structure of the updated model is similar to the original. If the first character is not in the set of variably assigned keys, the model proceeds according to the original algorithm. If the first character is in the set of variably assigned keys, the time is t_{min}; since the right thumb is used for the space key 70.49% of the time, we assume the left thumb is used for the first character 70.49% of the time for words beginning with a key from the variable set.

If both the current and previous keys may be pressed with either thumb, we assume expert users take a greedy approach to optimizing their thumb assignment plans. Like the original model, the total time the most recently used thumb is chosen is $T_{n-1} + t_{fitts}(key_{n-1}, key_n)$. Similarly, if the typist chooses the opposite thumb, the total time is $T_{n-k} + t_{fitts}(key_{n-k}, key_n)$. Since we assume a greedy approach, the typist will choose the shorter of these times:

$$min \begin{pmatrix} T_{n-1} + t_{fitts}(key_{n-1}, key_n), \\ T_{n-k} + t_{fitts}(key_{n-k}, key_n) \end{pmatrix} \quad (2)$$

However, as with the original implementation, each key will take at least t_{min} to depress. As a result, we use the larger of

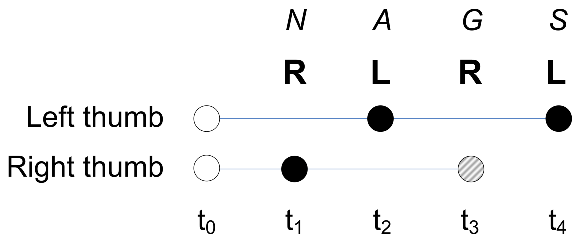

Figure 2. Updated model computation for the word *nags*.

t_{min} and the output from Eq. 2 as our final estimate:

$$T_n = max\left(\begin{array}{c} T_{n-1} + t_{min}, \\ min\left(\begin{array}{c} T_{n-1}+t_{fitts}(key_{n-1},key_n), \\ T_{n-k}+t_{fitts}(key_{n-k},key_n) \end{array}\right) \end{array}\right) \quad (3)$$

Updated Model Example

Consider the model's treatment of the word *nags* where the letter *g* is a variably assigned key which may be pressed by either thumb. Figure 2 is a visual representation of the key sequence where the total time to enter the word is given by T_4. Character entry is left-to-right; the lines represent the activities of the left and right thumbs. Dark circles represent statically assigned keys and the gray circle indicates a thumb usage choice made by the model. The first two letters are calculated as they would be with the original model. The letter *n* is pressed by the right thumb; the letter *a* is assigned to the left thumb, which moves from the space key it pressed at T_0. It may have reached the *a* key in the time it has taken the right thumb to press *n*, but the time increment should be at least t_{min}:

$$T_2 = max(T_1 + t_{min}, T_0 + t_{fitts}(< space >, a))$$

At this point the procedure diverges from the original model, in which the *g* key would have been statically assigned to the left thumb and the left thumb would move immediately from *a* to *g*, a time given by $T_2 + t_{fitts}(a, g)$. However, the right thumb has previously hit *n* which is close to *g*. In the updated model, we examine the possibility that using the right thumb might be faster. The time to move from *n* to *g* is $T_1 + t_{fitts}(n, g)$. Assuming the user adopts a greedy optimization strategy, she will choose the lesser of these two values, with the final time increment being at least t_{min}. Hence, from Eq. 3:

$$T_3 = max\left(\begin{array}{c} T_2 + t_{min}, \\ min\left(\begin{array}{c} T_2 + t_{fitts}(a, g), \\ T_1 + t_{fitts}(n, g) \end{array}\right) \end{array}\right)$$

We use the original model process to calculate the time for the final letter since the updated model has made the explicit choice to press the preceding key with the right thumb. Thus:

$$T_4 = max(T_3 + t_{min}, T_2 + t_{fitts}(a, s))$$

The speed increase suggested by the modified model is readily apparent. For the above example, the original model predicts an expert entry time of 1.11 seconds with *g* being statically assigned to the left thumb. In the updated model considering *g* as a variably assigned key predicts an expert entry time of 0.57 seconds.

Establishing Thumb-Key Assignments

The updated model as described is independent of the specific thumb–key assignments, so we must still determine which keys should be variably assigned. We collected no direct evidence of specific policies in our original experiments, so we instead examine the inter–key timing data for evidence of dual–thumb usage.

The existence of flexible key use is most likely to manifest itself in the data in the form of faster-than-expected inter–key timings. Specifically, time intervals between keys assigned to the same thumb that are consistently faster than Fitts' Law predicts are likely candidates. We measured this by comparing the mean deviation between letters typed by the same thumb to the predicted times. We computed 26 mean intervals (one for each letter) in this manner. The average deviation was 37 ms (σ = 41 ms); two letters had average deviations faster than one σ below the mean: *v* and *b*.

However, *v* is a significant outlier: transitions beginning with *v* were 187 ms faster than predicted, while the next lowest value was 97 ms (for *b*). The standard deviation excluding this outlier was σ' = 28 ms. The *g* and *y* keys (in addition *v* and *b*) were faster than σ' below the interval mean. We concluded from this analysis that a reasonable set of variably-assigned keys was *v*, *b*, *g* and *y*. The updated model predicts an expert speed for our Dell keyboard of **60.51** wpm (2% faster than the mean Dell session 20 rate) using these keys as our flexibly assigned set.

MODEL VALIDATION

The updated model prediction is not radically different from the original, but as we noted the original prediction masked lower-level inaccuracies. We validate the updated model via two methods. First, we use an error metric similar to the procedure discussed above, which compares the predicted inter–key transition times with the empirical data. Second, we validate the model using data from the second keyboard in our previous study (Targus brand). Since we have not examined the Targus data in the context of the theoretical model, it has not influenced the model's evolution and provides an unbiased data set against which to gauge the updated model's performance.

Model Error Metric

As before, we analyzed the timing data from the set of error–free sentences in sessions 16-20 where our participants had over 300 minutes of experience. Both the model and the empirical data yield a 26 × 26 matrix of transition times between each possible combination of any two key presses (the space key is excluded since the original model works at the word level). There are a total of 20,713 transitions, ranging from 1 (i→q) to 527 (t→h) occurrences.

To correct for sampling frequency, we weight individual cells in each matrix by their frequency of occurrence in the empirical data. We then compute an error metric by summing the squares of the differences between corresponding entries in the model and empirical matrices. Stated more formally, we weight a matrix of empirical transition times denoted $t_{[k_n,k_m]}$ and a matrix of modeled transition times $m_{[k_n,k_m]}$ using a matrix of transition frequencies $f_{[k_n,k_m]}$:

$$T_{[k_n,k_m]} = t_{[k_n,k_m]}/f_{[k_n,k_m]}$$

$$M_{[k_n,k_m]} = m_{[k_n,k_m]}/f_{[k_n,k_m]}$$

Our error metric E is then:

$$E = \sum_{i,j=0}^{n,m} (T_{[k_i,k_j]} - M_{[k_i,k_j]})^2 \quad (4)$$

We use this metric (in which lower is better) to evaluate the relative fitness of the model variations. The score for the original model is 37.1, while the the {v,b} and {v,b,g,y} versions of the updated model score 34.4 and 33.3, respectively: the latter—the final form of the updated model—is more than 10% improved over the original.

Targus Data

The original model assumes that each character is entered by a single key. However, the Targus keyboard used in our study has two space keys (Figure 1). Updating the model to account for both spaces is relatively straightforward and does not affect the model's overall structure—the model works at the word level and the space key is already a special case. Changing the model for the case of multiple keys for the same letter would require more extensive modifications.

The updated model predicts an expert rate of 60.62 wpm versus an original prediction of 57.95 wpm. Targus users, in comparison, had a mean session 20 rate of 58.74 wpm. The error metric score (Eq. 4) for the updated model is 47.67, 8% lower than the original score of 52.01.

FUTURE WORK

There are a number of avenues for future research in this area. Additional studies on mini-QWERTY keyboard use can help refine the model further and empirically examine factors like flexible key use. Examinations of more unique keyboard designs could test applicability of the model to less standard key layouts.

There is also ample opportunity for more basic Fitts' Law research. Although the standard Fitts model addresses large target regions and area cursor work [3] covers large selection regions, we are not aware of any work on their combination. Presumably, both non-trivial cursor (thumb) and target (key) widths may have some effect on motor performance, but research is needed to confirm or deny this supposition. Such work would have implications on the two–thumb model by providing more data on appropriate methods for calculating the effective key width. But this research also addresses more fundamental motor capabilities that are applicable to a wide variety of research (such as area cursors). Finally, we are also not aware of any research examining how aimed movement tasks are affected by crowded target environments. Whether Fitts' Law accurately models these situations or can be modified to do so is an open question, and one which has obvious implications for typing on miniature keyboards.

CONCLUSIONS

The original Mackenzie and Soukoreff two–thumb model provides a reasonable prediction of expert speed. However, our analysis of empirical data indicates that at a lower-level the model's predictions are consistently slower or faster for different classes of thumb transitions. This fact led us to re-examine the static thumb-key assignment assumption of Mackenzie and Soukoreff's original model and to extend its formulation to account for flexible key usage. This extension has its basis in both experimental data and anecdotal user reports. The altered model makes slightly faster wpm predictions with 8–10% lower error scores.

Analytic models like the two-thumb model can be powerful HCI tools: predictive statements about prospective designs have great utility. Comparing design alternatives, for example, can be done at very early stages of the design process using these models. Thus, their verification by empirical observation is a crucial bookend to their development. Apart from the improvements to the two-thumb model made in this work, the model's validation by our corpus of user data is a useful contribution to its development. As we have seen, the validation process not only provides an empirical basis for confidence in a model's predictions, but also can reveal opportunities for improvement and refinement.

ACKNOWLEDGMENTS

Thanks to Jim Foley and the Stephen J. Fleming Chair in Telecommunications for support of the first author. This work is funded in part by NSF Grant #0093291 and the Rehabilitation Engineering Research Center on Mobile Wireless Technologies for Persons with Disabilities, which is funded by the National Institute on Disability and Rehabilitation Research of the U.S. Department of Education grant #H133E010804.

REFERENCES

1. Clarkson, E., Clawson, J., Lyons, K., and Starner, T. An Empirical Study of Typing Rates on Mini–QWERTY Keyboards. In *CHI '05 Extended Abstracts*, pp. 1288–1291.

2. Fitts, P. M. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of Experimental Psychology*, 47:381–391, 1954.

3. Kabbash, P. and Buxton, W. A. S. The "Prince" Technique: Fitts' Law and Selection Using Area Cursors. In *Proceedings of CHI '95*, pp. 273–279.

4. MacKenzie, I. S. and Soukoreff, R. W. A Model of Two–Thumb Text Entry. In *Proceedings of Graphics Interface '02*, pp. 117–124.

5. MacKenzie, I. S. and Soukoreff, R. W. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts*, pp. 754–755.

6. Silfverberg, M., MacKenzie, I. S., and Korhonen, P. Predicting Text Entry Speed on Mobile Phones. In *Proceedings of CHI '00*, pp. 9–16.