

# Three Text Entry Methods Based on Smartphone Discrete Tilting: An Empirical Evaluation

Sandi Ljubic<sup>(✉)</sup>

Faculty of Engineering, University of Rijeka,  
Vukovarska 58, HR-51000 Rijeka, Croatia  
sandi.ljubic@riteh.hr

**Abstract.** Tilt can be used as an input modality for mobile devices, providing possibility for touch-free text entry implementations. This paper builds on previous research on modeling three text entry methods that utilize discrete tilt concept and rely on tilt-only interaction. These methods are analyzed a priori in theory, however, they have not yet been comparatively assessed by empirical approach. The work presented herein offers a contribution in that respect, as methods are evaluated against efficiency, learnability, and users' satisfaction in a study involving 20 participants. The efficiency of tilt-based text entry is inspected by observing performance metrics through repetition and practice time. Required physical and mental demands are also investigated, as well as are perceived frustration and overall effort. The results obtained from controlled text-entry experiment supported a detailed comparative analysis of methods' characteristics, and revealed a relation between theoretical predictions of upper-bound text entry speeds and real efficiency of the presented methods.

**Keywords:** Text entry · Tilt-based interaction · Mobile devices · Empirical evaluation

## 1 Background

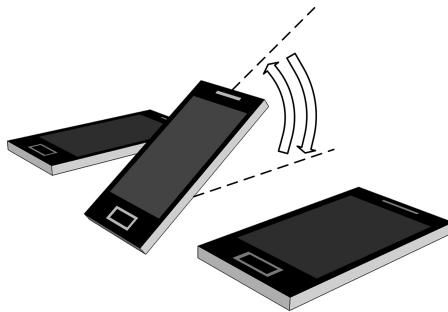
Dominant interaction techniques used for text entry across the most of the contemporary on-screen soft keyboards utilize direct touch (*tap*), sliding gestures (*swipe*), or a combination of both (*tap-and-swipe*). Apart from the touchscreen, motion sensors integrated into modern mobile devices can also be used for input control, providing a possibility to augment mobile text entry with new tilt-based methods.

Tilting has been initially analyzed as an input support for special hardware prototypes (*Unigesture* [1]), small watch-like devices (*TiltType* [2]), and feature phones with 12-button multitap-based keyboards (*TiltText* [3], *Vision TiltText* [4]). *Unigesture* approach prevents inputting individual letters, relying on an inference engine able to predict complete words based on the device tilting sequence. *TiltType* does not support single-handed text entry, and requires a combination of both button pressing and device tilting for character selection. *TiltText* technique allows for one-hand texting wherein tilt gestures are used to resolve character disambiguation after initial key press. *Vision TiltText* is functionally equivalent to the original *TiltText* method, but uses the built-in

camera for detecting both movement and tilt of the phone. None of the aforementioned methods operates using standard QWERTY, as zone-based character layouts are favored in those designs instead.

More recent motion-based solutions support text entry on present day mobile devices, namely touchscreen smartphones and tablets. *WalkType* [5] is an adaptive text entry system which uses accelerometer data to improve touch typing on a soft QWERTY keyboard, by compensating imprecise input while walking. *The Dasher* [6] is a text entry system in which a probabilistic predictive model and zoom-and-point interaction are simultaneously used for character selection. It supports multiple platforms and input modalities, here including continuous tilt as a way of pointing on a mobile device. Tilt-based target selection, enabled by continuous tilting of a tablet device, has been empirically investigated by Fitton et al. [7]. The proposed input technique was intended for text entry, however, experimental interface did not implement full-layout keyboard and procedure did not involve real text entry tasks.

Continuous tilting of a mobile device generally demands a high level of visual attention to select targets, i.e. characters accurately. The main emphasis of the author's previous research is put on a discrete tilt concept which, on the contrary, supports interaction less relying on visual feedback. Discrete tilt is an input primitive which is actually comprised of two movements: (i) device leaves the neutral position zone, rotating along the longitudinal or the lateral axis; (ii) after reaching a predefined threshold angle, device returns to the neutral position zone by an immediate backward movement. Altogether four basic input commands can be defined using discrete tilt: Roll Left, Roll Right, Pitch Down, and Pitch Up (depicted in Fig. 1).



**Fig. 1.** Pitch Up input command invoked by discrete tilt. The device needs to be twisted up-then-down against its lateral axis. Pitch Down is invoked in the same way, only down-then-up tilting sequence is used instead. A neutral position is assumed when the device is parallel to the horizontal plane.

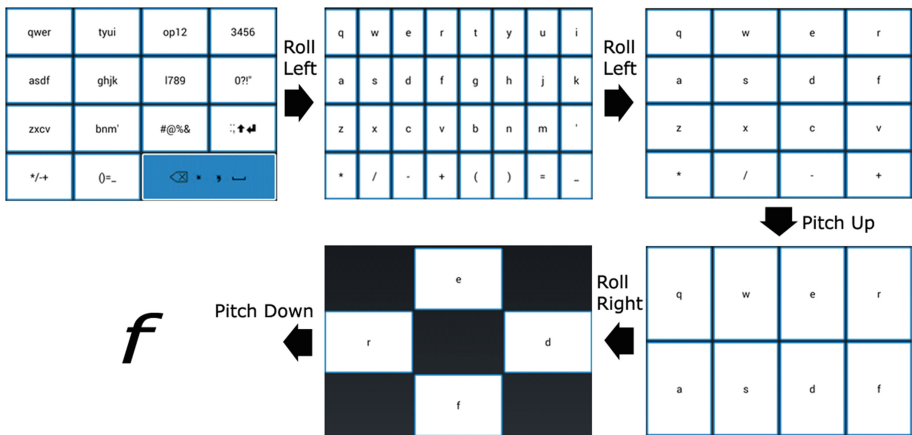
Discrete tilt concept is initially introduced in [8], along with the respective text entry method called *Keyboard Bisection* which uses an extended QWERTY-based layout and specially designed input scheme for character selection. This method subsequently motivated the implementation of two additional text entry solutions also relying on discrete tilt: *Single Cursor*, and *Quad Cursor* [9]. Basic concepts of the proposed methods are briefly described in the following.

**Keyboard Bisection (KB).** The tilt-based method in question uses discrete tilts for visual enlargement of a particular part of the keyboard. For example, Roll Left command will cause keyboard bisection resulting in the display of the left half of the current character layout. This way text entry is enabled with tilt movements exclusively, as particular character can be entered using step-by-step layout reduction (see example in Fig. 2). The KB method facilitates touch typing as well, since keyboard buttons are touch-enabled and their ever-increasing size allows for more precise selection. Nevertheless, in this paper the tilt-only interaction is of particular concern.

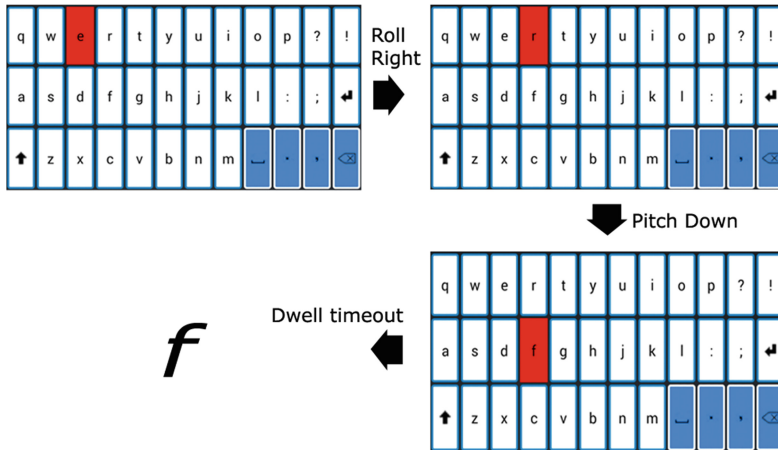
Buttons with different background color represent placeholders for frequently used characters/actions. These four common options can be alternatively selected using tilt-and-hold (or *long tilt*), a special interaction case when discrete tilt is extended by retaining the device position for 2 s before returning to the neutral position zone. Within the KB method, character selection involves exactly five discrete tilts, with the exception of shortcuts that can be selected either with one long tilt or four regular tilts.

**Single Cursor (SC).** This method abandons changeable layout concept and provides a consistent design with QWERTY alignment in three rows. It uses a specially visualized character (cursor) for marking the current position within the keyboard layout. Character selection is thus performed by tilting, i.e. moving the cursor in the appropriate direction. Input confirmation is achieved by dwelling in the neutral position zone (for a predefined amount of time) after successful discrete tilt execution. An example of character entry procedure using the SC method is shown in Fig. 3.

The efficiency of SC text entry method heavily depends on the cursor’s path between the initial position and the location of the following character/symbol. In that respect, the method supports circular navigation: a single discrete tilt can switch the cursor between the first and the twelfth column, as well as between the first and the third row. Additionally, long tilts can be used for shortcut activations, thus bypassing the need for constant cursor switching.

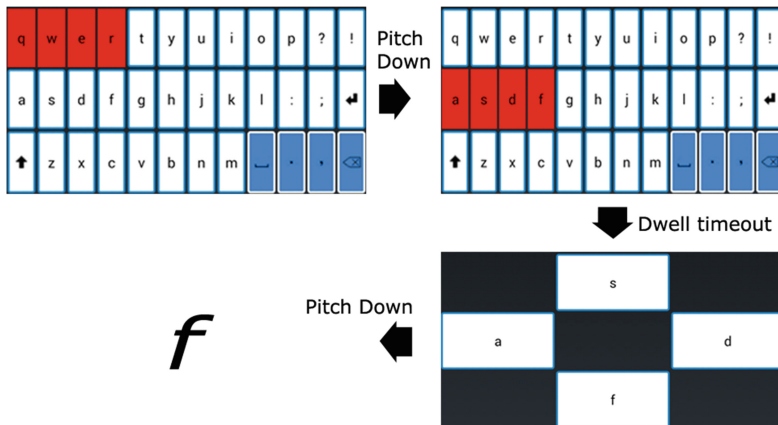


**Fig. 2.** Inputting character *f* by making use of tilt-only interaction within the *Keyboard Bisection* method. Different bisection strategies can be used for character selection.



**Fig. 3.** Inputting character *f* using *Single Cursor* method, assuming that cursor initially marks letter *e*.

**Quad Cursor (QC).** Finally, the QC method uses the same character layout as the SC method, but in addition virtually divided into quad-based zones. These zones are accessible via quad cursor, a group of four adjoining characters from the same row. Both tilting and dwelling are used for text entry, as shown in example in Fig. 4.



**Fig. 4.** Inputting character *f* using *Quad Cursor* method, assuming that cursor initially marks the letter group, i.e. the quad *q-w-e-r*.

The QC method requires three actions for character input: (i) navigating the cursor to the target quadruple, (ii) selecting the respective quadruple by dwelling, and (iii) making a final discrete tilt in order to choose among four presented characters. The third step is equal to the final letter resolving within the KB method. Circular navigation of quad cursor is allowed, as well as is shortcut activation using long tilts. While the QC method

supports faster positioning than the SC, it additionally requires one extra tilt for the final 4-letter disambiguation.

The three presented tilt-based methods are already analyzed using predictive modeling of upper-bound text entry speed [9]. Predictions are derived from a combination of a tilt-based movement model, and a linguistic model (digraph frequencies in English). Obtained results, presented in Table 1, refer to theoretical expert-level text entry performance. More details on the respective modeling procedure, results, and discussion can be found in author’s previous work [9].

**Table 1.** Upper-bound text entry speed predictions for presented tilt-based methods (*cf.* [9]). Predicted values hold for the maximum text entry expertise, as error-free input is assumed and all cognitive activities are ignored.

Text entry method	Text entry speed predictions (WPM <sub>max</sub> )	
	Long tilts not enabled	Using long tilts
Keyboard bisection	5.79	5.46
Single cursor	4.21	4.42
Quad cursor	5.16	5.15

Low text entry rates are predicted for discrete-tilt-based input methods. This is reasonable to expect, as the related input procedures assume several discrete tilts (and some dwell time) for a single character entry. Nevertheless, tilt-based input could provide support when typing on small screens becomes problematic, and/or in situations when visual contact with the smartphone display is obstructed. The presented methods can be furthermore enhanced using word prediction algorithms, but this research focuses solely on interaction aspects of tilt-based text entry. Real efficiency and workload demands of the presented methods were inspected by conducting a user study.

## 2 Empirical Evaluation: Participants, Apparatus, and Procedure

Twenty users were involved in text entry experiment (18 males, 2 females), their age ranging from 21 to 37 with an average of 25.25 years ( $SD = 5.17$ ). While all participants were regular users of a touchscreen smartphone (85 % were owners of an Android device), 18 of them had already been interacting with tilt-based mobile applications (mainly games). Users reported their preferred hands posture while holding smartphone as follows: 40 % for two-thumbs typing, another 40 % for single-handed usage, i.e. one-thumb typing in portrait orientation, and 20 % for cradling – a case where one hand is holding the device, while the other (usually the dominant one) performs the text entry. Only one participant was left-handed.

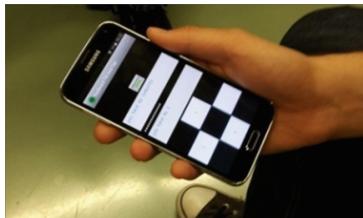
All three tilt-based text entry methods were tested on a Samsung Galaxy S5 smartphone (SM-G900F) running Android Lollipop OS. This device is  $142 \times 72.5 \times 8.1$  mm large and weighs 145 g. A simple Android application was developed, able to support tilt-based input methods, as well as to gather tilt actions, text entry events and the corresponding timing data. The application implements transcription-based text entry

tasks, meaning that each trial requires rewriting a displayed text phrase randomly selected from a 500 instances set developed by MacKenzie and Soukoreff [10]. All phrases in question consist exclusively of lowercase letters and space character, without any punctuation symbols. A single task was considered completed when a particular phrase was fully and correctly transcribed, so a distinct cognitive load for error checking was inherently involved. Data logging for a task instance began with the first discrete-tilt event and ended after entering the last correct character in a given phrase. All network-based services on the smartphone were turned off during the experiment. Regarding input methods' basic settings, threshold angles for pitch and roll movements were set to  $30^\circ$  and  $45^\circ$  respectively, while 1.2 s was assigned for dwell timeout. These values correspond to those used in modeling of upper-bound text entry speeds.

Before the testing, participants were informed about the main research goals. Participants' basic information about age, mobile device usage, and previous experience with tilt-based interaction was collected afterwards. This initial survey was followed by a detailed demonstration of text entry methods, in order to familiarize users with supported tilt-based character input schemes. Participants had no training sessions whatsoever. In the actual experiment, participants were instructed to enter three different text phrases, each one five times in a row, using available tilt-based methods (KB, SC, QC). The repetition part was applied in order to enhance users' skill acquisition, as well as to boost up the level of text entry performance. Single-handed interaction with the smartphone was obligatory, hence users had to hold the device in their dominant hand Fig. 5. Text entry tasks could have been accomplished while sitting or standing, so each user had to make a choice of respective position in regard to her/his own preference. Participants were furthermore instructed to input text "as quickly as possible, while trying to avoid errors", and to use long tilts for both space and backspace activations. Breaks were allowed between text entry tasks, as well as before shifting to another text entry method. A repeated measures (i.e. within-subjects) design was utilized, and the order of text entry methods was properly counterbalanced.

After the testing of each tilt-based text entry method, users were asked to estimate perceived workload by completing a questionnaire based on the rating part of the NASA-TLX (NASA Task Load Index). Namely, subjective opinions had to be reported on a 20-point Likert scales for five factors: mental demand, physical demand, frustration, performance, and effort.

Finally, at the end of the experiment, participants completed a short post-study survey, providing their concluding remarks on ease of use, perceived learnability, and overall satisfaction.



**Fig. 5.** A participant doing the text entry task using *Keyboard Bisection* method

### 3 Results and Discussion

Given that task iteration (of the same phrase) was not considered an independent variable in the experiment, the effect of repetitive entry was observed from the descriptive statistics standpoint only. Figure 6 presents achieved input performances averaged across five trials related to text entry iterations of a particular phrase. It can be seen that text entry performance generally improves with repetition, irrespective of the used tilt-based method. It is expected and understandable as task replication allowed for learning by means of inherently involved practice. Text entry speeds and total error rates are highly negatively correlated, which is a direct consequence of implemented tasks that required fully correct transcription.

Participants entered 900 phrases in total. After averaging data across unique phrases, altogether 180 text entry performance records were obtained: 20 participants  $\times$  3 methods  $\times$  3 unique phrases. Figure 7 shows levels of text entry performance, achieved using three tilt-based methods, for three different phrases. Unsurprisingly, performance enhancement through time can be observed once again.

To analyze the obtained data, a 3  $\times$  3 repeated measures ANOVA was used, with *Method* (KB, SC, QC) and *Phrase* (1st, 2nd, 3rd) being the within-subjects factors. The Greenhouse–Geisser  $\epsilon$  correction for the violation of sphericity was applied when appropriate. In cases where significant effect was found, post-hoc pairwise comparisons with Bonferroni adjustment were utilized. As text entry speed (i.e.WPM metric) is of particular concern in this paper, error rates are reported in graphs only.

The analysis revealed a significant effect of tilt-based entry *Method* on text entry speed ( $F_{1,208,22,956} = 87.647$ ,  $\epsilon = 0.604$ ,  $p < .001$ ). The effect of *Phrase* (i.e. practice through time) was also statistically significant ( $F_{1,391,26,437} = 51.833$ ,  $\epsilon = 0.696$ ,  $p < .001$ ). Finally, the effect of *Method\*Phrase* interaction was found statistically significant as well ( $F_{4,76} = 3.787$ ,  $p = .007$ ).

As for the pairwise comparisons, the differences between text entry methods, as well as between phrases, are reported in the following.

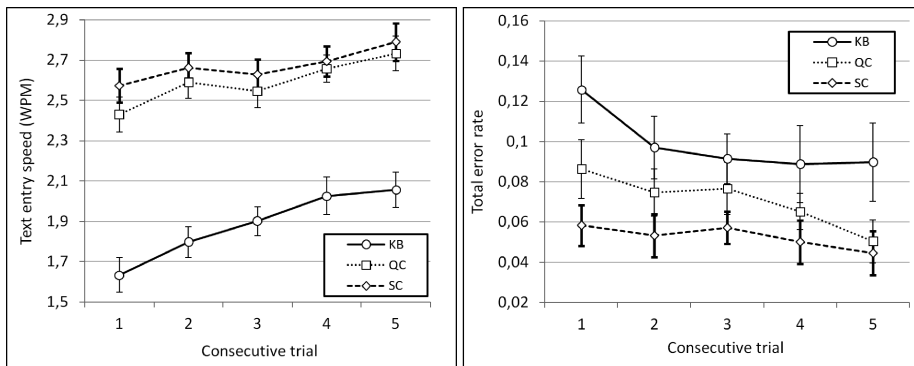
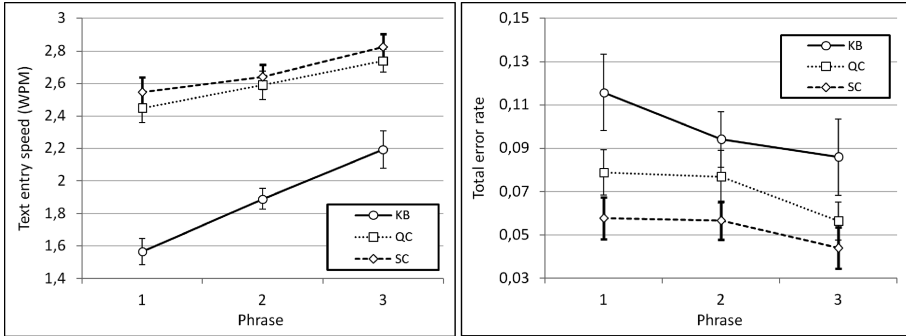


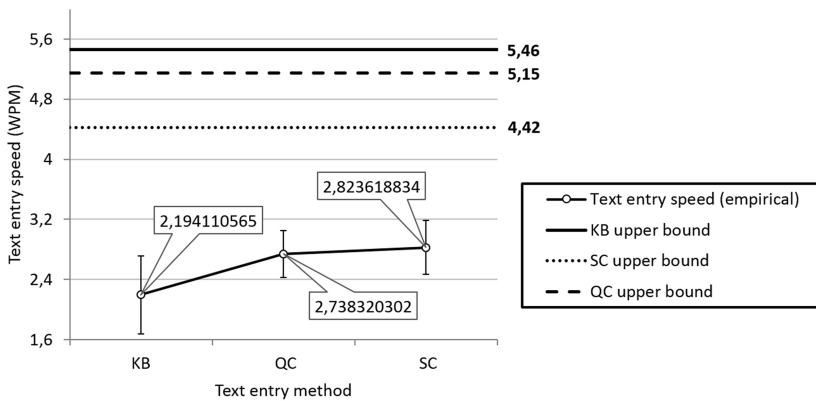
Fig. 6. Input performance averaged across five repetitive trials. The graphs show mean text entry rates and mean error rates, along with error bars with  $\pm 1$  standard error of the mean.



**Fig. 7.** Text entry performance averaged across three different phrases. The graphs show mean text entry rates and mean error rates, along with error bars with  $\pm 1$  standard error of the mean.

- SC vs. KB:  $(2.670 \pm 0.073 \text{ WPM})$  vs.  $(1.884 \pm 0.078 \text{ WPM})$ ,  $p < .001$
- QC vs. KB:  $(2.592 \pm 0.071 \text{ WPM})$  vs.  $(1.884 \pm 0.078 \text{ WPM})$ ,  $p < .001$
- SC vs. QC:  $(2.670 \pm 0.073 \text{ WPM})$  vs.  $(2.592 \pm 0.071 \text{ WPM})$ ,  $p = .886$ , ns
- Phrase 3 vs. Phrase 1:  $(2.585 \pm 0.075 \text{ WPM})$  vs.  $(2.187 \pm 0.064 \text{ WPM})$ ,  $p < .001$
- Phrase 3 vs. Phrase 2:  $(2.585 \pm 0.075 \text{ WPM})$  vs.  $(2.373 \pm 0.063 \text{ WPM})$ ,  $p = .001$
- Phrase 2 vs. Phrase 1:  $(2.373 \pm 0.063 \text{ WPM})$  vs.  $(2.187 \pm 0.064 \text{ WPM})$ ,  $p < .001$

The KB was the slowest of the three tilt-based text entry methods. The SC appeared to be the fastest one, with a less prominent difference when compared with the QC. Concerning text entry performance through time, participants achieved the best results when entering the third, i.e. the last phrase. In that respect, it was decided to use that level of text entry performance in the further investigation. Namely, the participants’ third-phrase performance was compared with theoretical predictions of upper-bound text entry speeds. The respective relations are shown in Fig. 8.



**Fig. 8.** The comparison of text entry speeds between empirical evaluation results and theoretical predictions. The graph shows mean text entry speeds and standard deviations, as well as upper-bound values (long tilts usage is assumed).



Interestingly, text entry speeds obtained from user testing are ordered just the opposite from what theoretical predictions suggest. While predictive models assume that bisection principle maintains the highest entry rate potential, in conducted experiment both the SC and the QC showed to be significantly faster than the KB method. In addition, real text entry speeds of the presented methods seem to be rather low when compared to their upper-bound limits. This discrepancy between theoretical predictions and empirical outcomes may raise the questions about validity of the modeling procedure. However, more detailed inspection of the obtained results can put a new light on the respective relation.

Upper-bound text entry speed predictions hold for total-expert behavior, which assumes all mental activities ignored and error making completely avoided. Experimental results revealed the largest error rate for the KB method (8.59 %), what makes accuracy improvement more promising for the KB than for the other two methods (5.64 % error rate with QC, 4.40 % for SC). It must be noted here that KB method implementation has one severe limitation in handling wrong bisections. Namely, bisection command has no undo option, thus causing errors to be even more time consuming. According to the aforementioned, error-free text entry would further benefit KB-method's performance the most.

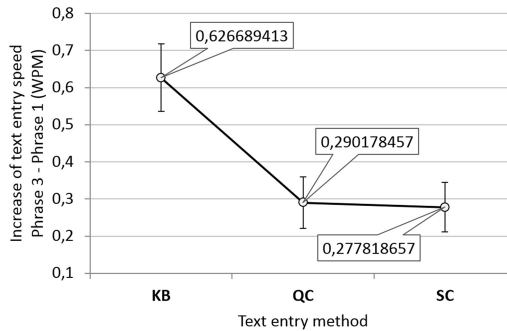
Regarding mental activities involved when using tilt-based methods, the KB seems to be more demanding because of its ever-changing appearance. This especially applies to novice users usually accustomed to positional consistency of keyboard characters. Reaching the text entry expert level with QC or SC requires "shortest path" continuous utilization when navigating cursor within an otherwise well-known character layout. On the other hand, expert usage of the KB method assumes mastering the bisection principle, i.e. learning and remembering character layouts invoked by particular bisection commands.

Given that the KB method offers more room for improvement, text entry speed can be expected to further increase at the greater pace for KB than for the other two methods. To corroborate this argument, efficiency improvement was analyzed basing on the difference between input speeds achieved while entering the first phrase (at the beginning of the experiment) and the third phrase (at the end of the experiment). The results are presented in Fig. 9.

Users improved their text entry speed over three phrases in such way that, in comparison with both the SC and the QC, KB improvement was more than twofold. The presented values correspond to 3.21 CPM (characters per minute) enhancement for KB, 1.57 CPM for QC, and 1.35 CPM for SC. Mean text entry speed enhancement differed significantly between observed methods ( $F_{2,38} = 6.524$ ,  $p < .05$ ). Post hoc pairwise comparisons confirmed statistical significance of the following differences:

- KB vs. QC:  $(0.627 \pm 0.091 \text{ WPM})$  vs.  $(0.290 \pm 0.070 \text{ WPM})$ ,  $p < .05$ ,
- KB vs. SC:  $(0.627 \pm 0.091 \text{ WPM})$  vs.  $(0.278 \pm 0.067 \text{ WPM})$ ,  $p < .05$ .

Text entry speeds converge to their upper bounds at different paces. If such improvement trends would hold for longer period, the relation between methods' real efficiencies would become in line with the ranking of the theoretical predictions. Predicted limits seem more convincing in that respect, regardless of initial divergence from empirical outcomes. Nevertheless, a more longitudinal study should be carried out to confirm such presumptions. To put things into perspective, it should be noted that



**Fig. 9.** An increase of text entry speed between entering the first phrase and the third phrase (mean values  $\pm 1$  standard error).

real text entry speeds were obtained from the experiment wherein users spent no more than 50 min per method. It is therefore reasonable to expect higher levels of text entry expertise on the longer run.

The qualitative evaluation of the presented methods was based on two questionnaires. The first one aimed for comparative rating of perceived workload, and was constructed using the “Raw TLX” format. The goal of the second one was to assess users’ final impressions on general usability attributes of the three different input techniques.

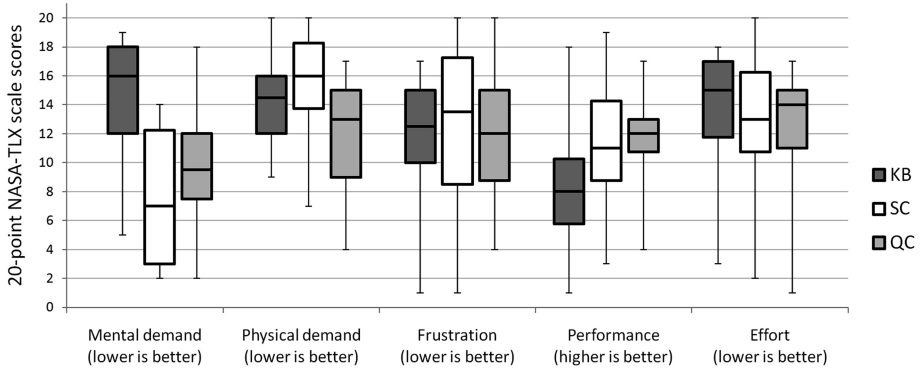
The box plots derived from the first questionnaire are shown in Fig. 10. The Friedman test was used to assess TLX-based scores. In cases where significant effect was confirmed, post hoc analysis was conducted by making use of Wilcoxon signed-rank tests with a Bonferroni correction applied (i.e. significance level set at  $p < .017$ ).

The type of text entry method being used had a significant effect on three factors: perceived mental demand ( $\chi^2(2) = 18.329$ ,  $p < .001$ ), physical demand ( $\chi^2(2) = 7.892$ ,  $p = .019$ ), and overall performance ( $\chi^2(2) = 8.778$ ,  $p = .012$ ). Statistically significant differences were not confirmed for perceived frustration ( $\chi^2(2) = 1.848$ ,  $p = .397$ ) and overall effort ( $\chi^2(2) = 5.688$ ,  $p = .058$ ). Post hoc analysis revealed the following facts:

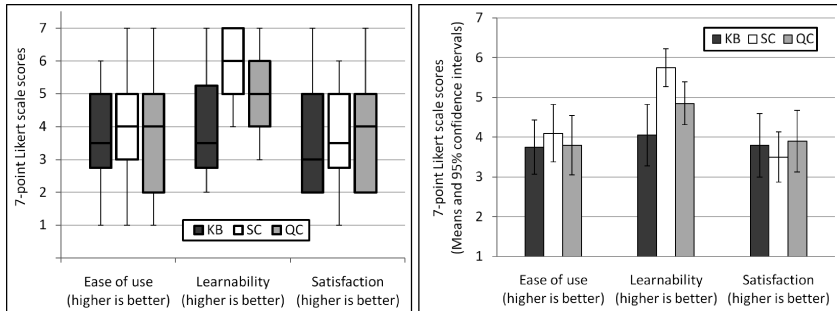
- The KB method required considerably higher mental activity than both the SC method ( $Z = -3.708$ ,  $p < .001$ ) and the QC method ( $Z = -3.247$ ,  $p = .001$ );
- Regarding physical activity, the SC method was significantly more demanding than the QC ( $Z = -2.468$ ,  $p = .014$ );
- Participants were more satisfied with their performance while using the QC than while using the KB ( $Z = -3.013$ ,  $p = .003$ ).

The workload assessment results confirmed the issues previously discussed. As opposed to cursor navigation concept, transformable character layout clearly imposed extra mental efforts. When it comes to physical demand within cursor-based methods, the SC involved much lengthier tilt-based distances between two characters, thus higher wrist fatigue in comparison with the QC is no surprise. The KB had the lowest level of perceived efficiency, which can in turn be contributed to the highest obtained error rate.

Ease of use, learnability, and overall satisfaction were the usability attributes inspected in the concluding survey. Participants rated three text entry methods against these attributes on a 7-point Likert scale. The results are shown in Fig. 11.



**Fig. 10.** Users' opinions on perceived workload of tilt-based input methods. For each factor, the corresponding box plots show minimum and maximum, 25 percentile (Q1), 75 percentile (Q3), and median value (M).



**Fig. 11.** Usability attributes: box plots (left), mean values and confidence intervals (right)

The methods were equally rated for ease of use, as well as for overall satisfaction. However, Friedman test revealed a statistically significant difference in perceived learnability ( $\chi^2(2) = 16.551, p < .001$ ). Post hoc Wilcoxon signed-rank tests confirmed that the SC method was the easiest to learn. From the perceived learnability standpoint, it significantly outperformed both the KB ( $Z = -2.932, p = .003$ ) and the QC ( $Z = -2.807, p = .005$ ). There was no significant difference in learnability between the KB and the QC ( $Z = -1.811, p = .07$ ). Cursor-based methods include a typical QWERTY layout and somewhat simpler input schemes than the mentally demanding KB method. In addition, the SC design is completely straightforward and does not involve any layout changes. In that respect, the obtained learnability ratings seem fully justified.

## 4 Conclusion

Three tilt-based methods that utilize discrete tilt concept and rely on tilt-only interaction were comparatively evaluated in a user study involving twenty participants. Empirically obtained text entry speeds were compared with their theoretical upper

bounds, previously derived by predictive modeling. The observed discrepancy between empirical results and theoretical predictions was discussed in detail. In addition, the results of qualitative assessment were presented, thus providing the insight into methods' workload demands and usability attributes.

In general, tilt-only interaction proved to be a viable option for text entry in the mobile domain. Input efficiency may not be as high as needed, but discrete-tilt concept could provide support in some specific use cases. Namely, tilt-based input offers a possibility for blind typing, as well as texting on particularly small devices where touch typing is unsuitable (e.g. smartwatches). As shown in this paper, a simpler character input scheme would be a better choice for securing initial acceptance.

## References

1. Sazawal, V., Want, R., Borriello, G.: The unigesture approach. In: Paternó, F. (ed.) *Mobile HCI 2002*. LNCS, vol. 2411, pp. 256–270. Springer, Heidelberg (2002)
2. Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G., Want, R.: TiltType: accelerometer-supported text entry for very small devices. In: *Proceedings of 15th ACM Symposium User Interface Software and Technology (UIST 2002)*, pp. 201–204. ACM Press, New York (2002)
3. Wigdor, D., Balakrishnan, R.: TiltText: Using tilt for text input to mobile phones. In: *Proceedings of 16th ACM Symposium User Interface Software and Technology (UIST 2003)*, pp. 81–90. ACM Press, New York (2003)
4. Wang, J., Zhai, S., Canny, J.: Camera phone based motion sensing: interaction techniques, applications and performance study. In: *Proceedings of 19th ACM Symposium User Interface Software and Technology (UIST 2006)*, pp. 101–110. ACM Press, New York (2006)
5. Goel, M., Findlater, L., Wobbrock, J.: WalkType: using accelerometer data to accommodate situational impairments in mobile touch screen text entry. In: *Proceedings of 2012 ACM Conference Human Factors in Computing Systems (CHI 2012)*, pp. 2687–2696. ACM Press, New York (2012)
6. Ward, D.J., Blackwell, A.F., MacKay, D.J.C.: Dasher – a data entry interface using continuous gestures and language models. In: *Proceedings of 13th ACM Symposium User Interface Software and Technology (UIST 2000)*, pp. 129–137. ACM Press, New York (2000)
7. Fitton, D., MacKenzie, I.S., Read, J.C., Horton, M.: Exploring tilt-based text input for mobile devices with teenagers. In: *Proceedings of 27th International BCS Human Computer Interaction Conference (BCS-HCI 2013)*, pp. 1–6. British Computer Society, London (2013)
8. Ljubic, S., Kukec, M., Glavinic, V.: Tilt-based support for multimodal text entry on touchscreen smartphones: using pitch and roll. In: Stephanidis, C., Antona, M. (eds.) *UAHCI 2013, Part III*. LNCS, vol. 8011, pp. 651–660. Springer, Heidelberg (2013)
9. Ljubic, S., Glavinic, V., Kukec, M.: Predicting upper-bound text entry speeds for discrete-tilt-based input on smartphones. *J. Interact. Sci.* **2**(3), 1–15 (2014)
10. MacKenzie, I.S., Soukoreff, R.W.: Phrase sets for evaluating text entry techniques. In: *Proceedings of Extended Abstracts on Human Factors in Computing Systems (CHI EA 2003)*, pp. 754–755. ACM Press, New York (2003)