Tilt-Based Support for Multimodal Text Entry on Touchscreen Smartphones: Using Pitch and Roll

Sandi Ljubic[1], Mihael Kukec[2], and Vlado Glavinic[3]

[1] Faculty of Engineering, University of Rijeka,
Vukovarska 58, 51000 Rijeka, Croatia
[2] Medimurje University of Applied Sciences in Cakovec,
Bana Josipa Jelacica 22a, 40000 Cakovec, Croatia
[3] Faculty of Electrical Engineering and Computing, University of Zagreb,
Unska 3, 10000 Zagreb, Croatia
sandi.ljubic@riteh.hr, mihael.kukec@mev.hr,
vlado.glavinic@fer.hr

Abstract. In this paper we propose a multimodal text entry method for touchscreen smartphones, where standard Tap modality can be used in combination with Pitch and Roll movements that change the orientation of the mobile device. Data from the built-in orientation sensors are used as a basis for commands that support character layout changing. Tilting the device in the appropriate direction will cause visual enlargement of the corresponding half of the current keyboard layout, thus enabling easier character selection, and solely sensor-based text entry. The prototype implementation of the proposed interaction method is analyzed and evaluated via usability testing experiments, with special focus on efficiency of text entry. As the proposed method is also applicable on touchscreen tablets, the form factor of mobile devices is reviewed with respect to text entry performance both of supported interaction modalities (tilt-only and tilt-and-tap) and of possible device orientations (portrait and landscape).

Keywords: text entry, multimodal interaction, accessibility, mobile device sensors, touchscreens.

1 Introduction

In the current mobile computing environment, the concept of quick message exchange has become a dominant phenomenon. Using standard messaging and social networks services "on the move" requires a convenient interaction with mobile devices and particularly efficient text entry methods. At the same time, touchscreens turned into the most popular platforms (including smartphones, tablets, and interactive tabletops) despite their chronic drawback: a generally slow, uncomfortable, and inaccurate typing [1]. On the other hand, touchscreen keyboards are software based, so they can support customization and automatic adaptation, as well as new interaction methods. The possibility of various soft keyboard implementations results with relevant

C. Stephanidis and M. Antona (Eds.): UAHCI/HCII 2013, Part III, LNCS 8011, pp. 651–660, 2013.

HCI-based research and experimental testing, with the main goal being the introduction of trouble-free and more efficient text entry. Dealing with typical problems such as the "fat-finger syndrome", optimal character layout, and size of keys involves investigations in linguistics, ergonomics, and mathematical optimization, while concepts of universal access [2] and universal usability [3] motivate design solutions for all users, including those with special needs.

Contemporary touchscreen keyboards are usually provided with both full QWERTY layout and integrated dictionary support with related algorithms able to predict the next letter in the word, and/or next word in the sentence that is being typed. Advanced devices additionally support text entry via voice input, but this functionality largely depends on voice processing capabilities and language models availability. When it comes to interaction, the dominant technique used for text entry across the most of the available keyboards consists of direct touch (*Tap*) and of sliding gesture (*Swipe*). While conventional keyboards usually use *Tap* for character entry, and *Swipe* for control features (e.g. character layout change, keyboard settings dialog activation), more sophisticated implementations require combined *Tap-and-Swipe* interaction. This especially applies for zone-based keyboards, where less frequently used letters are entered after tapping into the appropriate zone and swiping in the proper direction.

While quite a number of soft keyboards has been researched so far, tilt interaction modality is hardly ever considered to be a part of the final text entry solution for touchscreen smartphones and/or tablets. Our goal is to provide tilt-based support by introducing a text entry method which (i) runs on contemporary mobile devices with embedded motion sensors, (ii) supports multimodal interaction by enabling typing both with finger touch and wrist motion, and (iii) works with a QWERTY-based layout, hence ensuring a higher level of learnability.

2 Tilt Interaction with Mobile Devices: Related Work

Tilt and orientation of a mobile device have already been examined as a prospective mobile interaction technique [4, 5]. Early implementations provided possibilities for scrolling, changing screen orientation, and navigation through lists and menus, while text entry was addressed by *Unigesture* [6], *TiltType* [7], and *TiltText* [8] prototypes. However, *Unigesture* system relies on specially developed hardware, *TiltType* was designed for small watch-like devices and requires two-hand interaction, while the *TiltText* prototype supports multitap-based keyboards only. Recent research on this topic includes human factors in wrist-based input [9], comparative analysis of tilt, touch, speech, and foot interaction [10], and accelerometer-based text entry systems [11, 12]. *GesText* [11] represents a text entry design that relies solely on tilt input, with the respective prototype being based on the Wiimote device and a remote 32" screen. *WalkType* [12] uses accelerometer data to improve typing on a standard QWERTY layout, by compensating extraneous movement while walking. The *Dasher* project [13] provides a novel text entry method which discards the traditional keyboard concept, and introduces zoom-and-point interaction for selecting "flying letters" in 2D space. Related implementations are freely available, including smartphone applications with pointing gestures supported by tilt movements.

3 Tilt-Based Multimodal Text Entry Method

We selected Android as the development platform and utilized *Pitch* and *Roll* movements (around device's lateral and longitudinal axes) as valid tilt actions (Fig. 1).

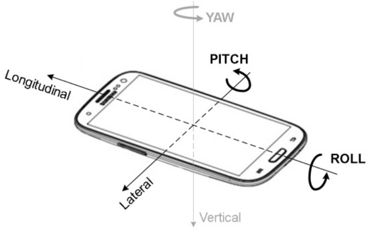

Fig. 1. Roll, *Pitch* and *Yaw* movements within orientation-based smartphone interaction. In our solution, tilting around the vertical axis is not interpreted for text entry support.

Altogether four tilt movements are mapped into the corresponding commands: twisting the smartphone up/down against the lateral axis generates *Pitch Down* and *Pitch Up*, while rotating left/right around the longitudinal axis results with *Roll Left* and *Roll Right* (see Fig. 2).

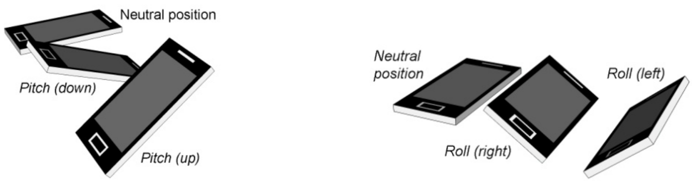

Fig. 2. Smartphone tilt movements used in our text entry method: *Pitch Down* (PD), *Pitch Up* (PU), *Roll Right* (RR), *Roll Left* (RL)

Fig. 3 shows the difference between a standard Android-based keyboard and our design, as well as a possible initial layout (*Level-1*) partition according to available tilt movements. It can be seen that the proposed keyboard configuration includes 15 interactive elements, each containing a set of four symbols for related characters and/or actions. While the QWERTY layout is available through the 8 upper left elements, the initial zone-based configuration additionally involves digits, punctuation marks, and supplementary symbols.

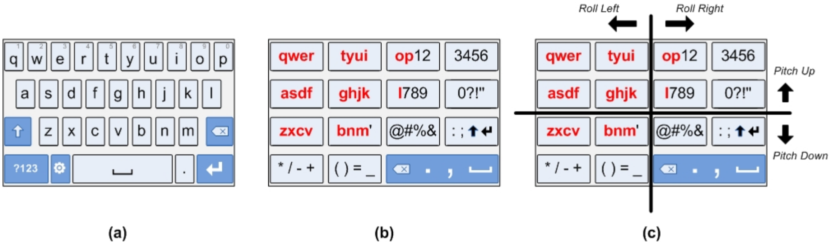

Fig. 3. Standard Android-based keyboard (a), proposed initial (*Level-1*) character layout for multimodal text entry (b), possible keyboard division at *Level-1* according to the available motion actions (c)

Character selection is enabled through direct touch from *Level-2* onwards, so two *Tap* actions are required for typing a particular letter. However, gradually reducing the available character set by smartphone tilting will make corresponding buttons larger, thus enabling each user to individually decide which level is most suitable for precise selection. Moreover, a character can be entered (or action fired) by using *Pitch* and *Roll* movements exclusively, as in case of *Level-5* final selection among maximum of four available options (Fig. 4).

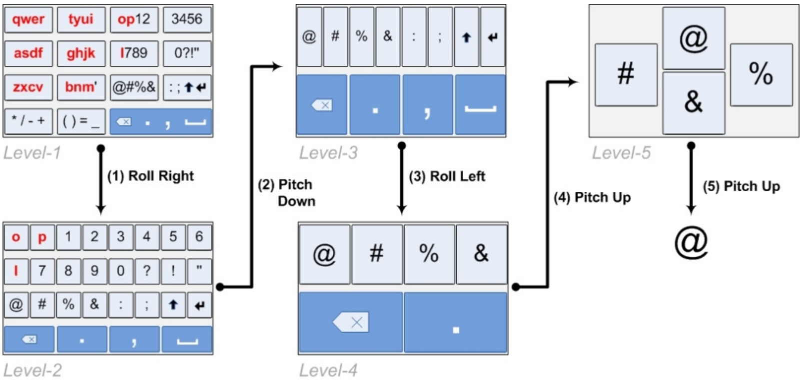

Fig. 4. Entering character "@" using *Pitch* and *Roll* only. User can also easily select "@" from *Level-2* onwards using *Tap*, with related key button dimensions being larger within each new level.

The largest interactive element in the initial layout contains symbols for frequently used characters/actions: backspace, full stop (period), comma, and space. Corresponding buttons are highlighted using different background color in order to notify the user on the possibility of shortcut activation. Namely, these four common options can be alternatively selected using tilt-and-hold, a special interaction case when *Pitch*/*Roll* is followed by retaining the device position for two seconds. Shortcut options can be selected regardless of the currently active character layout (*Level-1*–*Level-5*). Consequently, for our multimodal text entry method we define a set of available interactions *I*, with related events described in Table 1:

$$I = \{Tap, PD, PU, RL, RR, 2sPD, 2sPU, 2sRL, 2sRR\}.$$

Table 1. Interaction supported in proposed text entry method

Interaction	Triggered event
Tap [direct touch]	*Level-1*: displaying *Level-5* (4 symbols linked with corresponding button) *Level-2–Level-5*: character entry or action firing (uppercase, newline,…)
PD [*Pitch Down*]	*Level-1–Level-4*: reducing currently active character set to the bottom half *Level-5*: bottom-positioned character entry
PU [*Pitch Up*]	*Level-1–Level-4*: reducing currently active character set to the upper half *Level-5*: upper-positioned character entry
RL [*Roll Left*]	*Level-1–Level-4*: reducing currently active character set to the left half *Level-5*: left-positioned character entry
RR [*Roll Right*]	*Level-1–Level-4*: reducing currently active character set to the right half *Level-5*: right-positioned character entry
2sPD = *PD* + [2s hold]	*Level-1–Level-5*: comma character entry
2sPU = *PU* + [2s hold]	*Level-1–Level-5*: period character entry
2sRL = *RL* + [2s hold]	*Level-1–Level-5*: backspace (deleting last character)
2sRR = *RR* + [2s hold]	*Level-1–Level-5*: space (blank character entry)

4 Usability Evaluation

In order to evaluate the proposed method, we carried out usability testing. In the respective experiment, we wanted to analyze the effects on text entry performance (speed) and accuracy (rate of errors) of both device form factor and device orientation. In order to obtain qualitative results we focused on users' opinions regarding usability attributes and workload experienced with tilt-based interaction.

4.1 Materials, Methods, and Metrics

Participants. Twenty users were involved in our usability experiment (18 males, 2 females), their age ranging from 21 to 34 with an average of 25 years. While every user had previous experience working with touchscreen smartphones and tablets, 14 of them had already been interacting with tilt-based mobile applications (mainly games). The participants rated their text entry practice by approximation of the total number of text messages sent through various mobile messaging services on a daily basis. On average, this number appeared to be 10.

Apparatus. Four different mobile devices (D1–D4) were used by each participant during the experiment, two from the smartphone class, and two from the tablet one. All devices had a different form factor, according to the data presented in Table 2. In order to avoid a possible bias caused by devices' different technical specifications, the testing application was developed targeting minimal CPU and RAM requirements.

Table 2. Specifications of mobile device models used in usability testing experiment

Device	Model	Class	Display	Dimensions and Nominal Weight	CPU & RAM	OS
D1	*Samsung Galaxy Mini 2*	Smart-phone	**3.27"** 320x480 *TFT LCD capacitive*	*W*=58.6 mm *H*=109.4 mm *D*=11.6 mm ***NW*=105.3 g**	*ARM Cortex A5* 800 MHz 512 MB DDR	Android 2.3.6 *Gingerbread*
D2	*Samsung Galaxy S2*	Smart-phone	**4.3"** 480x800 *Amoled + capacitive*	*W*=66.1 mm *H*=125.3 mm *D*=8.5 mm ***NW*=116 g**	*DC ARM Cortex A9* 1.2 GHz 1 GB DDR	Android 2.3.4 *Gingerbread*
D3	*Huawei IDEOS S7 Slim*	Tablet	**7.0"** 480x800 *TFT LCD capacitive*	*W*=200 mm *H*=109.5 mm *D*=12.5 mm ***NW*=440 g**	*Scorpion* 1.0 GHz 512 MB DDR	Android 2.2 *Froyo*
D4	*Prestigio Multipad PMP7100C*	Tablet	**10.1"** 1024x600 *TFT LCD capacitive*	*W*=270 mm *H*=150 mm *D*=12 mm ***NW*=480 g**	*ARM Cortex A8* 1.0 GHz 256 MB DDR	Android 2.2 *Froyo*

Procedure. Before the actual experiment, we firstly collected participants' basic information about age, mobile devices' usage, and previous experience with tilt-based interaction. This initial survey was followed by a short individual practice session (approx. 30 minutes) for users to get familiar with both available devices and supported interaction techniques. Within the practice session, users were able to consider character layout changes and tilt angles used in the test application.

In the actual experiment, for each combination of available device (D1–D4) and device orientation (Portrait/Landscape), participants were instructed to enter three different text phrases using two interaction methods (tilt-only, tilt-and-tap). In landscape orientation, two-hand interaction with all devices was obligatory, while portrait orientation implied one-hand interaction only within the smartphone class. Each participant's 48 total phrases were randomly selected from a set of 500 phrases introduced in [14], all were in lowercase and without punctuation symbols. While users were instructed to input text "as quickly as possible, as accurately as possible", a single task was considered to be done when a particular phrase was fully and correctly transcribed. The input stream was analyzed "on the fly" with the test application itself, by monitoring phrase entry time and number of activated taps, tilts, and long tilts. To get around the possible learning effects in the experiment, we counterbalanced the sequence of experimental conditions using balanced Latin Squares. At the end of the experiment, users were asked to complete a post-study questionnaire, in order to get individual opinions about workload and usability attributes of the proposed method.

Metrics. Text entry speed was initially measured in characters per second, but is through a simple transformation here reported as words per minute (WPM). For accuracy metrics, we made use of the intensive work on text entry error rates provided in [15, 16, 17, 18]. Furthermore, we introduced a completely new metrics, which we devised specifically for our multimodal text entry method: *TiPC* (Tilts Per Character) and *TaPC* (Taps Per Character).

4.2 Results and Discussion

Participants entered 960 phrases in total: 480 for tilt-only tasks (where tapping was not allowed), and 480 for tilt-and-tap tasks (where interaction modality was the subject of free choice). The means and standard deviations for each condition and relevant metric are summarized in Table 3.

Table 3. Results: descriptive statistics summary

Modality	Device	Orient.	*WPM*		*Total Error rate*		*TiPC*		*TaPC*	
			Mean	SD	Mean	SD	Mean	SD	Mean	SD
Tilt-only	D1	P	1.600	0.526	0.052	0.069	4.974	0.726	No tapping allowed	
Tilt-only	D1	L	1.657	0.388	0.031	0.041	4.692	0.343		
Tilt-only	D2	P	1.442	0.422	0.058	0.060	5.028	0.646		
Tilt-only	D2	L	1.420	0.437	0.048	0.063	4.879	0.676		
Tilt-only	D3	P	1.603	0.402	0.042	0.042	4.760	0.358		
Tilt-only	D3	L	1.627	0.293	0.030	0.024	4.677	0.317		
Tilt-only	D4	P	1.372	0.297	0.032	0.042	4.627	0.314		
Tilt-only	D4	L	1.187	0.291	0.057	0.056	4.846	0.498		
T-a-T	D1	P	6.202	1.402	0.027	0.041	On the average, the total number of tilts represents only 0.29% of total taps count		2.098	0.180
T-a-T	D1	L	6.136	1.229	0.019	0.028			2.062	0.134
T-a-T	D2	P	6.185	1.367	0.018	0.024			2.072	0.109
T-a-T	D2	L	6.360	1.272	0.011	0.025			2.047	0.109
T-a-T	D3	P	5.593	1.252	0.028	0.045			2.112	0.220
T-a-T	D3	L	5.536	0.809	0.024	0.032			2.098	0.138
T-a-T	D4	P	5.656	1.320	0.022	0.038			2.073	0.189
T-a-T	D4	L	5.490	1.281	0.021	0.038			2.074	0.138

To analyze the obtained data, we ran a 4x2 repeated measures ANOVA on each metric, with Device (D1–D4) and Orientation (Portrait/Landscape) being the within-subjects factors. Interaction modality was not considered as a distinct factor, because there was no rationale for comparing completely different text entry strategies. In cases where significant effect was found, we utilized post-hoc pairwise comparisons with Bonferroni adjustment. The findings are presented in Table 4.

Device size/weight emerged as significant factor for both tilt-only and tilt-and-tap modality. Results showed that tilt-only text entry efficiency is higher when using smaller screens from both smartphone and tablet class. This can be explained by constraints of wrist movements (flexion, pronation, supination, ulnar and radial deviation) that are inherently higher when holding a larger mobile device. However, the source of no significant difference in performance between smaller smartphone (D1) and smaller tablet (D3) can be found in the nature of the interaction style: apparently, smaller smartphones are a better fit for one-hand tilt interaction, just as smaller tablets show to be a better choice for two-hand tilting.

Table 4. Significant effects and related post hoc analysis

Modality	Metric	Significant effect	Pairwise comparisons
Tilt-only	**WPM**	***Device*** $F(3, 57)=17.049$ $p<0.001$ $\eta p^2=0.473$	Post hoc analysis revealed significantly higher text entry performance when using D1 (WPM=1.629±0.095) over D2 (WPM=1.431±0.081; p=0.006) and D4 (WPM=1.280±0.060; p<0.001). Similarly, efficiency with D3 (WPM=1.615±0.070) outperforms significantly both D2 (p=0.006) and D4 (p<0.001).
T-a-T	**WPM**	***Device*** $F(3, 57)=6.833$ $p=0.001$ $\eta p^2=0.264$	Post hoc analysis revealed significantly higher text entry performance when using D1 (WPM=6.170±0.284) over D3 (WPM=5.565±0.212; p=0.026), as well as when using D2 (WPM=6.273±0.285) over both D3 (p=0.015) and D4 (WPM=5.573±0.284; p=0.004).

When it comes to tilt-and-tap, we must say that *Tap* was used as an arguably preferred option. In related tasks, tilts were used just occasionally for space or backspace actions. The reason for such phenomenon is rather obvious: there was no test user with any physical impairment, and given tasks requested from users to be as fast as possible. Since *Tap* is inherently faster than *Tilt*, all users decided to use it as the default input method. Results showed that tilt-and-tap (which came very closely to tap-only) text entry efficiency is significantly higher when using smartphones in relation to using tablets. Performance difference between smartphone D1 and tablet D4 however isn't statistically significant, but it is noteworthy nevertheless. Better tilt-and-tap efficiency with smartphones can be explained by a shorter hand/finger movement required for targeting each new character. Although tablet's keyboard presents larger buttons, its wider character layout does not help in improving speed of text entry, even in the case of two-hand interaction.

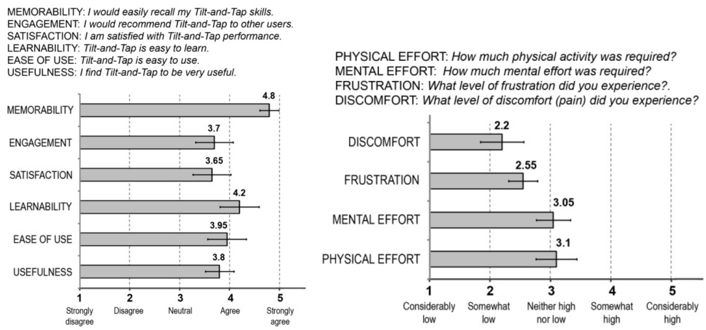

Fig. 5. Questionnaire response means: usability (left), and workload assessment (right)

No significant effects were found regarding *Total Error Rate*, *TiPC*, and *TaPC* metrics. *Total Error Rate* was reasonably low (maximum mean doesn't exceed 6%), while *TiPC* and *TaPC* mean values confirmed our expectations (close to 5, and slightly over 2, respectively). It is possible to predict *TiPC* and *TaPC* minimal values in advance, as they represent characteristics of related text entry techniques. In general, long (2s) tilts make 4.18% of tilt-only interaction, while wrong tilts (cases when wrong side of the keyboard was unintentionally activated) make only 1.4% of total tilts count. Wrong taps, cases when a wrong 4-key button was pressed at *Level-1* layout, make only 0.49% of total taps.

In the post-study questionnaire, usability attributes and workload were examined by 5-point Likert scale questions, with answers ranging from *Strongly disagree* (1) to *Strongly agree* (5) and *Considerably low* (1) to *Considerably high* (5), respectively. Users responded with encouraging outcomes (see Fig. 5).

5 Conclusion and Future Work

We have described a tilt-based multimodal text entry method for mobile devices that supports typing both with finger touch and wrist motion. Usability evaluation of both tilt-only and tilt-and-tap modality revealed smaller devices to be a better option when using the proposed method, and device orientation to be irrelevant for text entry performance. While the highest individual obtained WPM value for tilt-only was 2.63, even better performance could be expected after a longer period of usage, as users would eventually become more familiar with changing the keyboard layout. Enhancing the text entry speed with letter and/or word prediction algorithms was not the subject of our research, as we wanted to focus solely on functionality of multimodal interaction. In general, participants declared positive attitude towards method's usability, and considered physical and mental effort levels to be neither high nor low.

We believe that the proposed design could provide noteworthy in: (i) lowering the effect of the "fat-finger syndrome" thus improving text entry accuracy in mobile context, and (ii) assisting people who are unable to type conveniently on small screens. Although we were unable to recruit participants from the respective target group for usability testing, available results are promising and can validate our considerations.

Our future work plan includes providing alternative tilt-based methods where identical tilt actions will be used for shifting the cursor within a static keyboard layout, instead of constantly changing the character map. It would be very interesting to analyze a difference in text entry performance, as well as to find out whether users prefer a varying keyboard as presented in this paper, or the possibility to work with a consistent character layout when using the tilt-only entry method.

Acknowledgments. This paper describes the results of research being carried out within the project 036-0361994-1995 *Universal Middleware Platform for e-Learning Systems*, as well as within the program 036-1994 *Intelligent Support to Omnipresence of e-Learning Systems*, both funded by the Ministry of Science, Education and Sports of the Republic of Croatia.

References

1. Findlater, L., Wobbrock, J.O.: From Plastic to Pixels: In Search of Touch-Typing Touchscreen Keyboards. ACM Interactions XIX 3, 44–49 (2012)
2. Stephanidis, C.: Editorial. Universal Access in the Information Society 1(1), 1–3 (2001)
3. Shneiderman, B.: Universal Usability: Pushing Human-Computer Interaction Research to Empower Every Citizen. Communications ACM 43(5), 85–91 (2000)
4. Rekimoto, J.: Tilting Operations for Small Screen Interfaces. In: Proc. 9th ACM Symp. User Interface Software and Technology (UIST 1996), pp. 167–168. ACM Press, New York (1996)
5. Hinckley, K., Pierce, J., Sinclair, M., Horvitz, E.: Sensing Techniques for Mobile Interaction. In: Proc. 13th ACM Symp. User Interface Software and Technology (UIST 2000), pp. 91–100. ACM Press, New York (2000)
6. Sazawal, V., Want, R., Borriello, G.: The Unigesture Approach. In: Proc. 4th Int'l Symp. Mobile HCI (MobileHCI 2002), pp. 256–270. Springer, London (2002)
7. Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G., Want, R.: TiltType: Accelerometer-Supported Text Entry for Very Small Devices. In: Proc. 15th ACM Symp. User Interface Software and Technology (UIST 2002), pp. 201–204. ACM Press, New York (2002)
8. Wigdor, D., Balakrishnan, R.: TiltText: Using Tilt for Text Input to Mobile Phones. In: Proc. 16th ACM Symp. User Interface Software and Technology (UIST 2003), pp. 81–90. ACM Press, New York (2003)
9. Rahman, M., Gustafson, S., Irani, P., Subramanian, S.: Tilt Techniques: Investigating the Dexterity of Wrist-based Input. In: Proc. 27th Int'l Conf. Human Factors in Computing Systems (CHI 2009), pp. 1943–1952. ACM Press, New York (2009)
10. Dearman, D., Karlson, A., Meyers, B., Bederson, B.: Multi-Modal Text Entry and Selection on a Mobile Device. In: Proc. Graphics Interface 2010 (GI 2010), pp. 19–26. Canadian Information Processing Society, Toronto (2010)
11. Jones, E., Alexander, J., Andreou, A., Irani, P., Subramanian, S.: GesText: Accelerometer-Based Gestural Text-Entry Systems. In: Proc. 28th Int'l Conf. Human Factors in Computing Systems (CHI 2010), pp. 2173–2182. ACM Press, New York (2010)
12. Goel, M., Findlater, L., Wobbrock, J.: WalkType: Using Accelerometer Data to Accommodate Situational Impairments in Mobile Touch Screen Text Entry. In: Proc. 2012 ACM Conf. Human Factors in Computing Systems (CHI 2012), pp. 2687–2696. ACM Press, New York (2012)
13. Inference Group: The Dasher Project, `http://www.inference.phy.cam.ac.uk/dasher/`
14. MacKenzie, I.S., Soukoreff, R.W.: Phrase Sets for Evaluating Text Entry Techniques. In: Proc. Extended Abstracts on Human Factors in Computing Systems (CHI EA 2003), pp. 754–755. ACM Press, New York (2003)
15. Soukoreff, R.W., MacKenzie, I.S.: Measuring Errors in Text Entry Tasks: An Application of the Levenshtein String Distance Statistic. In: Proc. Extended Abstracts on Human Factors in Computing Systems (CHI EA 2001), pp. 319–320. ACM Press, New York (2001)
16. MacKenzie, I.S.: KSPC (Keystrokes per Character) as a Characteristic of Text Entry Techniques. In: Proc. 4th Int'l Symp. Mobile HCI (MobileHCI 2002), pp. 195–210. Springer, London (2002)
17. Soukoreff, R.W., MacKenzie, I.S.: Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric. In: Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI 2003), pp. 113–120. ACM Press, New York (2003)
18. Soukoreff, R.W., MacKenzie, I.S.: Recent Developments in Text-Entry Error Rate Measurement. In: Proc. Extended Abstracts on Human Factors in Computing Systems (CHI EA 2004), pp. 1425–1428. ACM Press, New York (2004)