MTIS: A Multi-Touch Text Input System

Michael Schmidt, Anja Fibich, and Gerhard Weber

Dresden University of Technology, Institute of Applied Science,
Human-Computer Interaction, Nöthnitzer Straße 46, 01062 Dresden
{Michael.Schmidt1,Anja.Fibich,Gerhard.Weber}@tu-dresden.de

Abstract. Entering text by gesture alphabets is not one of the most efficient methods. However, there are special applications and contexts where it shows advantages. Input with little focus of attention is possible and, for short phrases, transition to other input options may be more involving. The work at hand presents a new multi-touch gesture alphabet. Multi-touch can accelerate gesture input and provides the diversity that allows to confine to single strokes that demand less attention. We analyzed the characteristics of the alphabet and compared it to a single-touch variant. Detailed investigations of text input by gestures and results of a user study are provided. The investigations revealed preferences of users and showed the need for individualization and self-definition of gestures. To meet this demands, our approach for classifying template defined letters is demonstrated.

Keywords: gesture alphabet, text input, classification, recognition, template-based, multi-touch.

1 Introduction and Motivation

Applications for controlling UI by symbolic surface gestures are browsers, air traffic control [3], accessibility of mobile [16,8] or special [20] devices or in sketching tools, directly [1,23], or to support sketch recognition [11,4]. Symbolic gestures are applied in several text entry systems, too. Gesture alphabets are, for instance, 'EdgeWrite' [22], 'Minimal Device Independent Text Input Method' (MDITIM) [7], or the 'Graffiti' alphabet realized in the Palm OS. Such methods can offer advantages over conventional text input methods. As an abstraction to handwriting, such symbols are easier to recognize by pattern matching methods. Additionally, they allow for shorthand writing and demand less attention by writing letters on top of each other. Though efficiency of these input methods is low compared to, for instance, virtual keyboards, users may still prefer them [9]. We introduce a new gesture alphabet consisting of multi-touch symbols. That allows for greater variability and similarity to Latin letters while improving writing speed. The multi-touch gesture recognizer utilized is capable of recognizing all types of symbolic surface gestures and can be trained by templates. This provides the option to compare other gesture alphabets directly. Furthermore, the classifier detects gestures invariant to rotation, scaling and speed. Therefore, it scales with users' experience in text input and is suitable for mobile devices.

N. Streitz and C. Stephanidis (Eds.): DAPI/HCII 2013, LNCS 8028, pp. 62–71, 2013.

2 State of the Art

For text input in mobile environments different approaches exist. Most common is the 12-key keyboard, capable to achieve around 10 words per minute (with T9 support, approximately 20 wpm are possible) [13]. In substitution to QUERTY hardware-keyboards, less space demanding virtual keyboards are common for smartphones, too. Besides typing, other silent input systems include handwriting and entering symbols of gesture alphabets. While handwriting is natural to most users, it is still challenging for pattern recognizers. Virtual keyboard input is fast (conservative model estimation predicts at least 28 wpm [24]), but unnatural and prone to parallax errors.

Several approaches improve on pure keyboard input and diminish differences between gesturing and typing. One way is to re-arrange or group symbols to provide structures for a short way hierarchical targeting by gestural movements [17,14,15]. In [18], letters are split into a set of abstract and recurring segments to define a hierarchical structure whose navigation produces them. Other methods support gesturing on virtual keyboards to connect keys to words. Though performance gains by gesturing on virtual keyboards are possible in general, there is much room for improvements due to special layouts [19]. In [2], such input can be done even bi-manually. By interpretation of the drawn shape on the basis of a large dictionary, the approach in [10] requires no precise targeting of keys.

All these methods lack the ability of eyesfree writing. Although, methods that interpret the shape - as in [10] - demand less attention from users as they are allowed to trade accuracy for speed [13]. Overall, the precise targeting necessary for input per soft keyboards, hierarchical selection and handwriting requires focusing the input area. Eyesfree entry allows to spend more attention to supporting input techniques as word completion [12].

Pure gesturing is an alternative that uses an input area more efficiently by writing letters on top of each other. In [6], this concept was introduced as heads-up writing and it was stated being easier on the wrist for users when compared to handwriting. Another advantage of writing discrete symbols in single movements and an unrelated manner is the possibility of entering text with less attention to the screen or even without visual feedback. In [13], the term ‘focus of attention’ (FOA) is introduced. It indicates the level of attention a task requires from a user. For instance, blind writing of memorized text has a lower FOA than writing it with a text entry method that requires additional observation of input [13].

Gesture alphabets discretize letters by variable degrees of abstraction. The more abstract, the more robust is recognition, though users’ ability to grasp them intuitively may suffer. ‘EdgeWrite’ [22] and MDITIM [7] facilitate robust recognition by utilizing sequences of only a few directional strokes. This enables both methods to be less dependent of input devices and input by tracking of eye movements, for instance. Little more resemblance to Latin letters can be found in ‘Unistrokes’ [6]. The ‘Graffiti’ alphabet, in contrast, mimics most Latin letters and is therefore more intuitive. Its successor ‘Graffiti 2’, though prone to input errors (19% at approximately 12% due to classification errors), compared to virtual keyboard is preferred by users (with word completion) because it is seen more intuitive and usable as well as less exhausting [9]. This is interesting regarding that a slow input rate (9 wpm), approximately two-thirds the input speed shown on virtual keyboard, was observed.

3 Multi-Touch Text Input

In the following, we use the same terms and taxonomy as in [21] (see Fig. 1). A gesture consists of at least one stroke, i.e., a phase during which at least one contact touches the sensing device. Each stroke can have one or multiple (partly) concurrent touches. 'Single-touch' and 'multi-touch' refers to gestures consisting of one stroke with one or multiple touches, respectively. A 'multi-stroke' is characterized by more than one subsequent but no concurrent touches. A gesture of multiple strokes containing one being multi-touch is referred to as 'sequential multi-touch'.

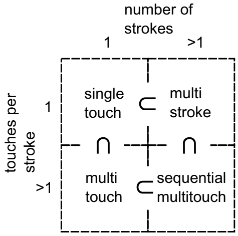

Fig. 1. The taxonomy of surface gestures from [21]. Gestures are classified by their number of strokes and the maximum number of concurrent touches within these strokes.

We present a multi-touch gesture alphabet based on the two 'Graffiti' versions. Our aim is to speed up writing by avoiding multi-stroke symbols and introducing multi-touch symbols for complex letters. The level of abstraction is low compared to 'Unistrokes' and most symbols resemble upper case Latin block letters. We balance efficiency and intuition by abstracting letters that are drawn multi-stroke when writing with a pen or that have a shape that can be produced faster using multiple touches. Variations of several letters are supported to allow for further writing speedup or individual style. In our user study, we investigated the advantages and drawbacks of this approach. For this purpose two alphabets were created, a single-stroke reference alphabet and the multi-touch text input method. Figure 2 illustrates the two alphabets. On the left side, the single-stroke reference alphabet is shown. It consists of letters from 'Graffiti' 1+2, stripped by multi-stroke symbols and enlarged by few simplified alternatives for some letters. On the right side, our multi-touch alphabet is demonstrated. We replaced selected symbols by multi-touch counterparts. The remaining single-stroke gestures are shared for the same letters between both alphabets.

To recognize both gesture alphabets of Fig. 2, the gesture classifier presented in [21] was used. It recognizes arbitrary surface gestures that were trained by

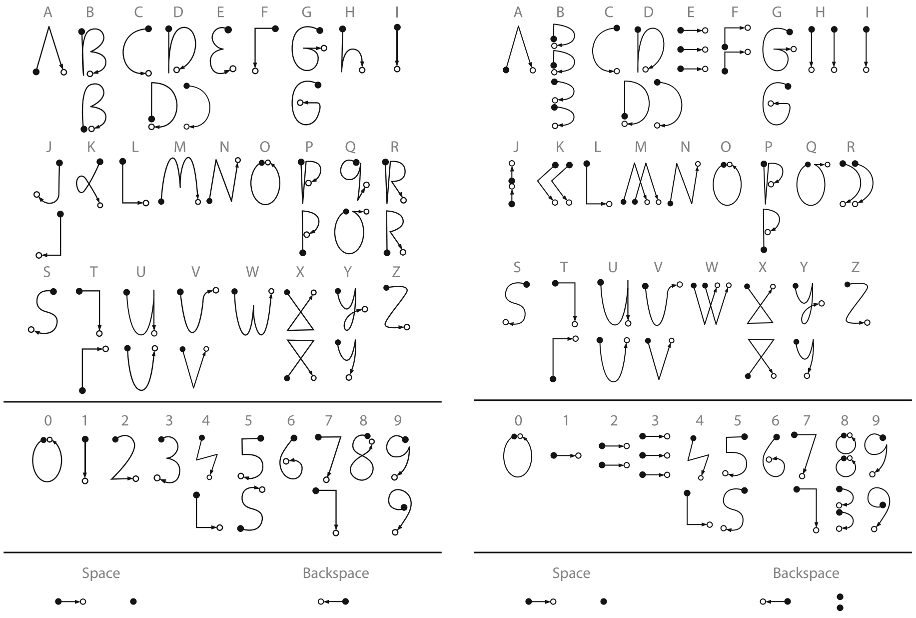

Fig. 2. Illustration of input for the single-touch (left) and the multi-touch alphabet (right). Symbols are drawn in one stroke, but may consist of up to three movements of simultaneous touches (only on the right side). A black dot depicts a touch, the arrow its movement and the blank circle the position where the contact is lost.

templates. This way a fair comparison is not hindered by different classification methods. Two gesture templates per symbol were defined in advance by the authors.

4 Gesture Recognition

Details of the applied on-line gesture recognition routine are found in [21]. The approach is based on statistical (Bayesian) classification and comprises comparisons by features of the shape and structural properties of a gesture's segments (tokens). The segmentation process splits gestures by their trajectories of touches on the sensing device. The pseudo code adapted from [21] is given in Algorithm 1. The procedure 'CompareGestures' computes the similarity of an input I and a template gesture T with the same number of tokens n.[1]

Algorithm 1 finds the best bijective matching between two sets of tokens by the negated sum of their pairwise distances of normalized structural features and their shape. Structural features of a token are its relative position, size, slope and rotation within the gesture. Shapes are compared by the Procrustes

---

[1] An input is not allowed to match a template with a different number of tokens.

Algorithm 1. CompareGestures(I, T, n)

```
▷ INPUT: T - template gesture; I - input gesture; n - token count
for all $i = 1 \to n$ do
    $\boldsymbol{y}_T^{(i)} \leftarrow$ ExtractStructuralFeatures(T,i)
    $\boldsymbol{z}_T^{(i)} \leftarrow$ GetTrajectory(T,i)
    $\boldsymbol{y}_I^{(i)} \leftarrow$ ExtractStructuralFeatures(I,i)
    $\boldsymbol{z}_I^{(i)} \leftarrow$ GetTrajectory(I,i)
end for
if $n == 1$ then
    return ProcrustesShapeDistance($\boldsymbol{z}_T^{(1)}$,$\boldsymbol{z}_I^{(1)}$)
else
    for all $i = 1 \to n$ do
        for all $j = 1 \to n$ do
            md $\leftarrow$ SquaredEuclideanDistance($\boldsymbol{y}_T^{(i)}$,$\boldsymbol{y}_I^{(j)}$)
            sd $\leftarrow$ ProcrustesShapeDistance($\boldsymbol{z}_T^{(i)}$,$\boldsymbol{z}_I^{(j)}$)
            mm[i,j] $\leftarrow -md - sd$
        end for
    end for
    return MaximumMatchingValue(mm)
end if
```

analysis. It performs normalization of the trajectories to match them in sample rate, size and rotation[2]. A squared Euclidean distance of thus normalized 'shape signatures' is computed. The interested reader is referred to [21] for a detailed description of features and routines. Possible improvements of the classification routine concerning parameter estimation to account for feature correlations are discussed there as well. Rejection of an input is possible by defining a distance threshold that shall not be exceeded. We did not apply rejection for MTIS.

5 Evaluation

A first version of our multi-touch alphabet was evaluated with 20 users in [5]. Organized in a between group design, the multi-touch alphabet was compared to the 'Graffiti 2' alphabet by performing text-copy tasks[3]. In a questionnaire, users declared the multi-touch gesture alphabet more usable in the whole and in detail conducive to learning. Furthermore, the input of multi-touch symbols was faster in the learning phase and significant faster during tests, when averaged over the same test sentences.[4] However, 'Graffiti 2' contains few multi-stroke symbols that require a time-out for proper recognition. To gain better comparability,

---

[2] We restricted rotational invariance to 20 degrees to include gestures differing in this aspect mostly.

[3] Text-copy tasks are preferred over text-creation tasks to minimize mental workload and errors of incorrect memorization or spelling[13].

[4] In detail, letters 'E','H','M','W', for instance, proved to be significant faster.

both alphabets were modified. We restrict all symbols to single strokes in this work. Within the multi-touch alphabet, former single-touch symbols of the letters 'R' and 'K' were replaced by abstract multi-touch symbols[5].

We were interested in further insights of gesture text entry methods, more specifically, in the following questions:

- Differ the two alphabets regarding intuitivity, error rate and satisfaction?
- Compared to conservative methods, is text input per gestures perceived as useful?
- Are abstract symbols preferred, or the ones more similar to handwriting?
- Would users like to specify their own gestures?

The classifier allows for template-based specification of symbols and users can individualize their alphabets. We were keen to know if users accept such possibilities to enhance their text input system.

The second evaluation was done again with an SMS writing tool developed in [5]. This tool can be used to define and teach gesture alphabets and presents text for testing purposes. In the same between group design, each group was confronted with a different gesture alphabet and the following test schedule:

- Introduction to the alphabet, its notation and the evaluation routine.
- Training:
    1. Visual templates of symbols/variants were retraced by the user.
    2. Aiding icons were shown and symbols entered twice altogether in random order. Gestures entered correctly were automatically added as templates to make classification more robust to between user variations.[6]
    3. Symbols were entered (random) without visual aid and in the users' preferred variant.
- Test: Users were requested to enter text presented to them as quickly and accurately as possible. Corrections were done by backspace buttons or gestures only. A reminder of the gesture alphabet could be displayed on press of a help button. Altogether, 16 phrases per participant were entered by gestures.
    1. A pangram[7] was written per virtual keyboard by the user.
    2. Sentences (letters only and letters plus numbers) were entered per gestures.
    3. The pangram was entered again, this time per gestures.
- Survey per questionnaire on perceived usability and subjective preferences. A free text field gave the option to provide additional comments.

---

[5] Pretests indicated they were hard to remember, but wanted to know if abstraction or intuition is preferred.

[6] This individualization without knowledge of the users has its drawbacks. The templates did not always represent the users' style. Instructed not to play around, some users did so to test recognition. Others drew the gestures still very meticulous.

[7] Lower case version of German phrase 'Franz jagt im komplett verwahrlosten Taxi quer durch Bayern'.

As for the first evaluation [5], we used a convenience sample of participants. Of the 12 new participants (7 female, 5 male, aged 19-34), nine were students or former students of computer science. Most stated to write approximately 1-5 SMS per day. One each answered to this question with 0, 6-10, 11-20. Six participants use virtual keyboard when writing SMS, four the 12-key keyboard and two use the ‘QWERTZ’ hardware keyboard.

6 Results

Considering characters per minute (cpm, as defined in [13]) when writing the pangram, the multi-touch group (M=36.93, SD = 4.63, N=6) was significantly faster (two-sample t-test; t=3.1, DF = 10, p = 0.01) than the single-touch group (M = 28.13, SD = 5.19, N=6).[8] Considering overall input during the test phase, strong tendency to faster multi-touch input is shown by the same test (t = 2.08, p = 0.06).

Cumulative error rates (containing all misinterpreted input) during the third phase of training (without aid) were 13.85 for multi-touch and at 17.44 for single-touch. This difference showed no significance (two-sample t-test).

In our survey, we collected answers to questions regarding subjective sensations of error rate, temporal demand, stress and ease of memorization when writing per gestures. On a five point Likert scale (1 worst - 5 best) all items were rated slightly higher for multi-touch and in average between 3 and 4. Highest difference (3.3 versus 4) is shown in subjective error rate of gesture recognition.

The direct comparison of input by gesture alphabets to the usual text input method showed preference of the latter. In both groups, most answers to preferred technique, fun, error rate, stress and concentration got average values of 3 to 3.83 on the five point Likert scales. Increasing values describe tendencies in favor of the usual text input. Differences between the two groups are negligibly small (0.16 or less) and insignificant. One exception occurred for fun, which got a better average rating of 2 for multi-touch compared to 3 for single-touch.

All participants were confronted with a picture of both alphabets and a short explanation of the differences in the final question of the interview. Until this time, it was not communicated that a comparative evaluation took place. Asked what alphabet they would prefer, given only this choice, three (two within the multi-touch group) selected the multi-touch version and nine the single-touch alphabet.

Figure 3 displays the distribution of answers, when users were asked on preferred variants of input possibilities.

Besides evident tendencies towards specific versions of ‘T’, ‘X’ and indicated preferences for ‘J’, ‘V’, ‘Y’, most symbols - including commands for space and backspace - are not assigned to a generally favored gesture. This impression is affirmed by our logging data that additionally provides information for numbers. Number ‘8’ (multi-touch) and ‘9’ were favored in their figurative variations

---

[8] For comparison, input by virtual keyboard achieved in average 96.69 cpm.

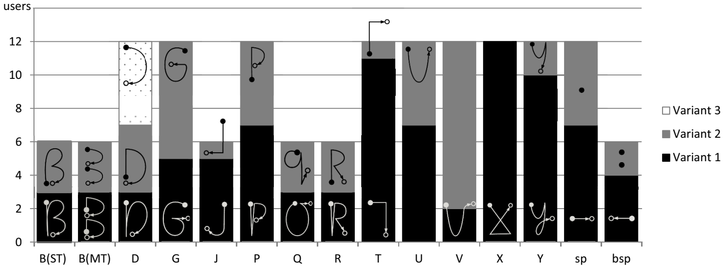

Fig. 3. Numbers of users that prefer one of the provided input methods for different symbols. Note that not all symbols are available within both alphabets.

whereas number '4' was entered in one third of the time in the abstract variant, too. For spaces and backspaces the relations were as indicated in Fig. 3.

7 Discussion and Outlook

Our results affirmed that, besides additional input options, a performance gain can be achieved when multi-touch is involved. The multi-touch alphabet itself, however, is in need of improvement. In spite of better ratings, users tend not to select it when given the choice between the single-touch and the multi-touch alphabet. Reasons for this can be found in users' comments. In three comments of multi-touch users, the awkward input for 'R' was criticized and in two that of 'K'. For both letters the help was called most often, too. Further discontents were communicated regarding the horizontal direction of the numbers 1-3 and the frequent changes in drawing directions. One participant wished for a mixture of both alphabets.

The distribution of selected input styles within the questionnaire suggest the development of gesture alphabets that allow for much more variation in users' input than it is provided in available tools. We conclude the need of individualization when offering input by gesture alphabets. This deduction is supported by answers in our questionnaire concerning whether users would rather like to specify their own gestures. All but one participant responded positively.

The instruments to evolve arbitrary gesture alphabets (see Fig. 1) by continuous modifications were presented. The template-based classification can be used for fair evaluations of different gesture alphabets under the same conditions. Rates of classification errors should be adequate for that purpose. Memorization performance and learning curves are features to be evaluated thoroughly in the future. However, a usable text input system including adaptability is still to be built. Ambiguous definitions should be prevented and aided recall created dynamically from specified templates. To be of real use, it would require sophisticated help and text correction as it is common for other input methods. Still, applications of gesture alphabets fall into a niche. Investigations of how and in which contexts this input option is of advantage are to be done. We imagine a text input system on mobile phones for the blind or visually impaired users. Possibly, with concepts of [20], a training mechanism (not necessarily per haptic sensations) can be developed that allows for learning and recalling gestures as a blind user. Input by a self defined alphabet may be fast and useful for this user group.

Acknowledgments. The authors kindly thank all participants of our tests for their help and constructive input.

References

1. Apte, A., Vo, V., Kimura, T.D.: Recognizing multistroke geometric shapes: an experimental evaluation. In: UIST 1993: Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology, pp. 121–128. ACM, New York (1993)
2. Bi, X., Chelba, C., Ouyang, T., Partridge, K., Zhai, S.: Bimanual gesture keyboard. In: Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology, UIST 2012, pp. 137–146. ACM, New York (2012)
3. Chatty, S., Lecoanet, P.: Pen computing for air traffic control. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Common Ground, CHI 1996, pp. 87–94. ACM, New York (1996)
4. Coyette, A., Schimke, S., Vanderdonckt, J., Vielhauer, C.: Trainable sketch recognizer for graphical user interface design. In: Baranauskas, C., Abascal, J., Barbosa, S.D.J. (eds.) INTERACT 2007. LNCS, vol. 4662, pp. 124–135. Springer, Heidelberg (2007)
5. Fibich, A.: Evaluation von Gestenalphabeten. Diploma thesis, Dresden University of Technology (2012)
6. Goldberg, D., Richardson, C.: Touch-typing with a stylus. In: Proceedings of the INTERACT 1993 and CHI 1993 Conference on Human Factors in Computing Systems, CHI 1993, pp. 80–87. ACM, New York (1993)
7. Isokoski, P., Raisamo, R.: Device independent text input: a rationale and an example. In: Proceedings of the Working Conference on Advanced Visual Interfaces, AVI 2000, pp. 76–83. ACM, New York (2000)
8. Kane, S.K., Bigham, J.P., Wobbrock, J.O.: Slide rule: making mobile touch screens accessible to blind people using multi-touch interaction techniques. In: Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility, Assets 2008, pp. 73–80. ACM, New York (2008)
9. Költringer, T., Grechenig, T.: Comparing the immediate usability of graffiti 2 and virtual keyboard. In: CHI 2004 Extended Abstracts on Human Factors in Computing Systems, CHI EA 2004, pp. 1175–1178. ACM, New York (2004)
10. Kristensson, P.-O., Zhai, S.: Shark2: a large vocabulary shorthand writing system for pen-based computers. In: Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology, UIST 2004, pp. 43–52. ACM, New York (2004)
11. Landay, J.A., Myers, B.A.: Interactive sketching for the early stages of user interface design. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1995, pp. 43–50. ACM Press/Addison-Wesley Publishing Co., New York (1995)

12. Scott MacKenzie, I., Chen, J., Oniszczak, A.: Unipad: single stroke text entry with language-based acceleration. In: Proceedings of the 4th Nordic Conference on Human-Computer Interaction: Changing Roles, NordiCHI 2006, pp. 78–85. ACM, New York (2006)
13. MacKenzie, I.S., Tanaka-Ishii, K.: Text Entry Systems: Mobility, Accessibility, Universality. The Morgan Kaufmann Series in Interactive Technologies, Boston (2007)
14. Mankoff, J., Abowd, G.D.: Cirrin: a word-level unistroke keyboard for pen input. In: Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology, UIST 1998, pp. 213–214. ACM, New York (1998)
15. Martin, B.: Virhkey: a virtual hyperbolic keyboard with gesture interaction and visual feedback for mobile devices. In: Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services, MobileHCI 2005, pp. 99–106. ACM, New York (2005)
16. McGookin, D., Brewster, S., Jiang, W.: Investigating touchscreen accessibility for people with visual impairments. In: Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges, NordiCHI 2008, pp. 298–307. ACM, New York (2008)
17. Perlin, K.: Quikwriting: continuous stylus-based text entry. In: Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology, UIST 1998, pp. 215–216. ACM, New York (1998)
18. Poirier, F., Belatar, M.: Glyph 2: une saisie de texte avec deux appuis de touche par caractère - principes et comparaisons. In: Proceedings of the 18th International Conferenceof the Association Francophone d'Interaction Homme-Machine, IHM 2006, pp. 159–162. ACM, New York (2006)
19. Rick, J.: Performance optimizations of virtual keyboards for stroke-based text entry on a touch-based tabletop. In: Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology, UIST 2010, pp. 77–86. ACM, New York (2010)
20. Schmidt, M., Weber, G.: Multitouch Haptic Interaction. In: Stephanidis, C. (ed.) UAHCI 2009, Part II. LNCS, vol. 5615, pp. 574–582. Springer, Heidelberg (2009)
21. Schmidt, M., Weber, G.: Template based classification of multi-touch gestures. Pattern Recognition (2013), doi:10.1016/j.patcog.2013.02.001
22. Wobbrock, J.O., Hudson, S.E., Mankoff, J., Simpson, R.C.: Edgewrite: A versatile design for text entry and control. Technical report (2006)
23. Yu, B., Cai, S.: A domain-independent system for sketch recognition. In: Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, GRAPHITE 2003, pp. 141–146. ACM, New York (2003)
24. Zhai, S., Hunter, M., Smith, B.A.: Performance optimization of virtual keyboards. Human-Computer Interaction, 89–129 (2002)