

# One-Dimensional Handwriting: Inputting Letters and Words on Smart Glasses

Chun Yu Ke Sun Mingyuan Zhong Xincheng Li Peijun Zhao Yuanchun Shi

Key Laboratory of Pervasive Computing, Ministry of Education

Tsinghua National Laboratory for Information Science and Technology

Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China  
{chunyu, shiyc}@tsinghua.edu.cn, {k-sun14, zhongmy14, lixc12, zhaoy13}@mails.tsinghua.edu.cn

## ABSTRACT

We present 1D Handwriting, a unistroke gesture technique enabling text entry on a one-dimensional interface. The challenge is to map two-dimensional handwriting to a reduced one-dimensional space, while achieving a balance between memorability and performance efficiency. After an iterative design, we finally derive a set of ambiguous two-length unistroke gestures, each mapping to 1-4 letters. To input words, we design a Bayesian algorithm that takes into account the probability of gestures and the language model. To input letters, we design a pause gesture allowing users to switch into letter selection mode seamlessly. Users studies show that 1D Handwriting significantly outperforms a selection-based technique (a variation of 1Line Keyboard) for both letter input (4.67 WPM vs. 4.20 WPM) and word input (9.72 WPM vs. 8.10 WPM). With extensive training, text entry rate can reach 19.6 WPM. Users' subjective feedback indicates 1D Handwriting is easy to learn and efficient to use. Moreover, it has several potential applications for other one-dimensional constrained interfaces.

## Author Keywords

Text entry; one-dimensional input; unistroke gestures.

## ACM Classification Keywords

H5.2. Information interfaces and presentation: User interfaces — Input devices and strategies.

## INTRODUCTION

Nowadays, many new smart personal devices are emerging. Such devices typically have a constrained input interface due to the limited size of the form factor. Consequently, text entry is difficult on these devices, which prohibits their broader use. In this paper, we focus on one-dimensional text entry for devices that have only one-dimensional input

signals, or the input capacity on one dimension is much greater than that on the other. Examples include the touchable spectacle frame of a smart glass (e.g. Google Glass), the edge of a side screen of a smart phone (e.g. GALAXY Note Edge), and a smart wristband.

Researchers have proposed various techniques to enable text entry on constrained interfaces (e.g. a keypad, a joystick, a tracking ball, and air interaction etc.). These techniques include coding letters [16], unistroke gestures [2], ambiguous keyboards [18], and gesture keyboards [12, 31]. However, most of these techniques are for two-dimensional constrained interfaces or interfaces with a limited number of buttons or gestures.

In this paper, we present 1D Handwriting, which enables users to perform unistroke gestures on a one-dimensional interface to input text. We research, implement and evaluate 1D Handwriting with Google Glass, a representative one-dimensional input device. The biggest challenge encountered is that many letters look similar, if not identical, after projected into a reduced one-dimensional space. Therefore, our goal is to derive a set of one-dimensional gestures that strikes a balance between memorability, input accuracy and input speed.

After a careful and iterative design, our result is a set of ambiguous unistroke handwriting gestures based on sub-stroke of two-level lengths. To complete the input, we design a one-dimensional gesture recognition algorithm that classifies input strokes in a probabilistic way. For letter input, we design a smooth transition gesture allowing the user to first gesture a stroke (representing a group of letters) and then select the target letter (within the group) without lifting his/her finger. For word input, we use a Bayesian approach that takes both the probability of gesture and the language model into account to interpret users' input.

To evaluate the performance of 1D Handwriting, we conduct three lab experiments. Results show that 1D Handwriting provides immediate usability: With limited training, users can achieve 4.67 WPM for letter level input (error rate = 0.78%) and 9.72 WPM for word level input (error rate = 0.25%). These speeds are comparable to those proposed for two-dimensional constrained interfaces, and significantly outperforms the one-dimensional selection-based technique (the Baseline in our experiments) by 11.4%

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

CHI'16, May 07-12, 2016, San Jose, CA, USA

© 2016 ACM. ISBN 978-1-4503-3362-7/16/05...\$15.00

DOI: <http://dx.doi.org/10.1145/2858036.2858542>

for letter input and 20.0% for word input. With extensive training focused on a single phrase (6 words, 14 distinct letters, 7 distinct gestures), text entry rate can reach 19.6 WPM after the tenth input. Users’ subjective feedbacks indicate that 1D Handwriting is fun to use and its stroke gestures are easy to remember. In the future, we plan to apply 1D Handwriting to other one-dimensional constrained interfaces.

In the remainder of this paper, we first review previous work relevant to our research. We then introduce the design process for deriving the 1D Handwriting gestures. We then report three user studies that examine the performance of 1D Handwriting for inputting letters and words. We finally conclude our research with discussion of limitations and future search directions.

**RELATED WORK**

**Letter input on constrained interfaces**

H4-Writer [16] uses four buttons to input letters (base 4), and leverages Huffman coding to minimize key sequences by considering letter frequency. A 20.4 WPM text entry rate could be achieved after ten 35-50 minutes practice sessions. Meanwhile, a gesture-based version of H4-Writer [3] could achieve 5.3-6.6 WPM using a touchpad and a Wiimote.

Multi-step interaction is another means for inputting letters on constrained user interface. GesText [11] allows users to input letters (e.g. with a Wiimote) with two-step directional gestures (5.4 WPM) (Figure 1, left and middle). LURD-Writer [7] uses horizontal and vertical movements of cursor to recursively reduce the letter range to the target one (8 WPM) (Figure 1, right). Swipeboard [6] divides the QWERTY keyboard on a very small touchscreen into nine regions. A first swipe specifies the region, and a second specifies the character within that region.

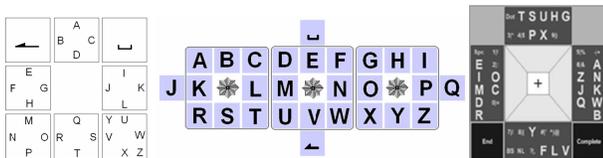


Figure 1. Matrix-based layout and tri-center layout of GesText (left and middle) [11]; LURD-Writer (right) [7]

Unistroke letters were first proposed for pen-based input. Figure 2 shows some sample letters of two well-known unistroke vocabularies. *Unistrokes* [8] was designed to be fast to write, and less prone to recognition errors. *Graffiti* [2] was designed to resemble handwritten Roman. Researchers showed users could learn Graffiti in minutes [15]; but after extensive training (twenty fifteen-phrase sessions), *Unistrokes* could be significantly faster (15.8 WPM vs. 11.4 WPM) [5]. Unistroke letters could also be used for eyes-free input on mobile devices [21] and for gesturing in the air [17].

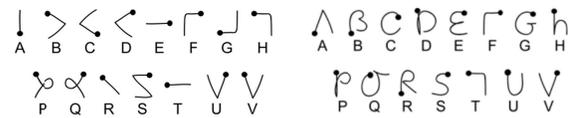


Figure 2. Sample gesture of Unistrokes (left) and Graffiti (right) [5]

EdgeWrite [28] was originally proposed by Wobbrock et al. to enable unistroke input while providing high accuracy and motion stability for individuals with motor impairments. The first version fixed a square hole over a touchscreen where users traversed the edges and diagonals to input text (Figure 3). Gesture recognition was performed based on the sequence of corners that were hit. EdgeWrite’s alphabet was designed to mimic its handwritten counterparts for quick learning. Edgwrite could also be adapted to other input devices such as joysticks and touchpads [26, 30], Trackball [24], and spatial keys [29].

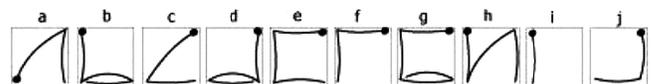


Figure 3. Sample gestures of Edgwrite [28]

*UniGest* [4] defined a gesture alphabet based on Wiimote [23], where each gesture was designed to have at most two primitive motions. Writing-with-joystick [10] defined gestures based on on-boundary and off-boundary strokes. Evaluation results showed text entry rate achieved 4.5 WPM after a brief learning session. For both techniques, visual similarity to handwritten letters was the guideline for designing the gestures of individual letters.

**Word input on constrained interfaces**

Ambiguous keyboards are widely used when the screen size is too small to accommodate all letters or only a limited number of signals (e.g. keys or gestures) are available due to constraints from both the device and/or user. The idea is to overload individual keys with more than one letter (e.g. T9) and resolve word ambiguity by matching key sequence to possible words in a predefined vocabulary. The mapping of letters to keys can be done in different ways [18].

Gesture keyboards [12, 31] enable users to input a word using a single unistroke gesture. The algorithm interprets input using both language and gesture information. A special advantage of gesture keyboard is that users do not have to lift up the pen or finger between letters within one word. Therefore, a gesture keyboard is suitable for mid-air interaction (e.g. Vulture [17]) where inputting a delimiter is difficult. MotionInput [19] realizes an ambiguous gesture keyboard that arranges eight ambiguous keys into a circle. Users traverse the keys to input words with a Leap Motion controller.

One-dimensional text entry techniques already exist; all are selection-based. The 1Line Keyboard [13] was proposed to reduce the size of the screen occupied by the keyboard. It reduces the QWERTY keyboard into one row containing

eight keys, with each key overloaded with multiple letters. The Minuum keyboard [21], based on 1Line Keyboard, enables text entry on Google Glass. However, there is only a demo video describing it; no implementation details nor performance data are available. Circle keyboard [20] organizes letters into a circle in alphabetical order. Users rotate a Wiimote to select target letters. The text entry rate was 10.2 WPM. Walmsley et al. [22] applied the source-channel approach to Circle keyboard to allow fast and imprecise input. Experiment results showed expert users could reach 21WPM after 6.7 hours practicing.

In sum, an input interface can be constraint in terms of the number of input signals (keys and usable gestures) or the number of dimensions of the input space. In this work, we target the constraint interface where touch can be performed only in one-dimensional space. Besides, we focus on handwriting approaches rather than selection-based ones for text entry.

### STUDY 1: DESIGN OF 1D HANDWRITING GESTURES

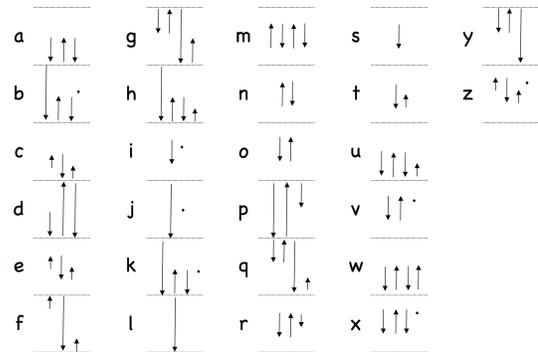
The goal was to gain an understanding about user acceptance of 1D Handwriting gestures as well as the input capacity of Google Glass, the device we used to experiment our idea in this research. To achieve this, we first derived a set of 1D Handwriting stroke gestures, and then asked individual participant to assess their intuitiveness by performing them on Google Glass. Based on the subjective and objective results, we agreed on the design guidelines for the final handwriting gestures, and revised our design.

#### Phase 1 —Initial design of handwriting gestures

In our first trial, we adopted a user-participatory approach [25]. We recruited eight participants and asked them to design a set of one-dimensional gestures for Google Glass. The process lasted for one hour for each participant to design gestures for all twenty-six letters of the alphabet. In the end, four participants proposed a code-based design, while the other four proposed a handwriting design. Unfortunately, none of the obtained gesture sets was satisfactory, because they were not memorable. This result revealed the difficulty of designing appropriate 1D Handwriting stroke gestures, especially within a very limited time. However, according to the post-experiment interviews, participants (including those who proposed the coding-based design) consistently agreed that the handwriting design was much more recognizable and memorable than the code-based design.

We then decided to design a set of handwriting stroke gestures by ourselves. The guideline was to design a 1D Handwriting stroke gesture for each letter to best mimic their two-dimensional counterparts. Figure 4 summarizes the handwriting gestures. For some letters, the design was easy and intuitive. For example, the handwriting stroke gesture of “w” should be a four-stroke gesture “back—forward—back—forward”. However, for others it was not an easy task since several letters were similar, if not identical, when projected into a reduced one-dimensional

space. In such cases, we designed strokes with more details (e.g. adding a short stroke at the end to distinguish “q” from “y”), or use a dot to distinguish them, such as “o” and “v”, “b” and “k”, “i” and “s”, and “j” and “l”. The design of handwriting gestures employed strokes of three levels of lengths (i.e. short, medium and long) and a dot.



**Figure 4. Handwriting gestures used in Study 1. Dots (“.”) were included in this design. However, these were consequently dropped in favor of faster input speed. Arrows represent the directions for each sub-stroke in a gesture. They are drawn continuously in a gesture.**

#### Phase 2 — User assessment and data collection

##### Apparatus

We used Google Glass for this experiment. As shown in Figure 5, there is a touchpad mounted on the right spectacle frame. The touchpad (7.0 centimeters long and 0.8 centimeters wide) supports up to three touch points with an input resolution of 1366×160. Although Google Glass is not a strict 1D input device, the vertical space is very limited for input. In addition, there is a virtual screen (resolution: 640×360) and a built-in speaker on the device. The Google Glass system was XE 22 running on Android 4.4.4. We developed a software that recorded all touch events, including time and coordinate information.



**Figure 5. The touchpad configuration of Google Glass**

##### Design and procedure

Twelve participants (six males and six females; Age: 18-25) were recruited from Tsinghua University campus. None had previously used a Google Glass. The experiment took about one hour to complete. At first, each participant was asked to memorize the handwriting gesture set (Figure 4). Then, they were instructed to perform these gestures (five times per letter) on Google Glass at their comfortable speed. For reference purposes, a sheet containing every letter-gesture pair was placed beside participants in case they forgot a gesture during the experiment. At the end of the

experiment, participants rated the perceived memorability of each individual gesture on a 5-point Likert scale. In total, we collected 1560 handwriting gestures for 26 letters.

**Results**

*Memorability.* Although the overall handwriting strokes resembled two-dimensional handwriting, the small adjustments (i.e. short strokes and dots) made it difficult for users to remember actual strokes. According to participants’ comments, the small adjustments seemed to be defined in an arbitrary way in that no consistent rule could be drawn out. Participants had to remember the strokes by heart. Unfortunately, this was not easy given the limited time for learning the stroke gestures. The fact that participants constantly referred to the gesture sheet also confirmed this difficulty. Moreover, participants’ subjective assessment on memorability was diverse, even for the same handwriting stroke gestures, (e.g. “q”, “h”, “g” etc.). This suggested individual participants should have discrepant models for interpreting the handwriting stroke gestures.

*Input accuracy.* It was difficult for participants to distinguish between three levels of stroke length. Figure 6 shows the distribution of short, medium and long sub-strokes after being normalized to the length of the longest sub-stroke within a gesture. There was much overlap between the three-level lengths of sub-strokes. In addition, participants occasionally performed unintended hooks at the beginning or end of a stroke, making short strokes even more difficult to recognize by the algorithm. Therefore, we intend to drop short strokes from our design and be in favor of a two-level design of stroke length (medium and long).

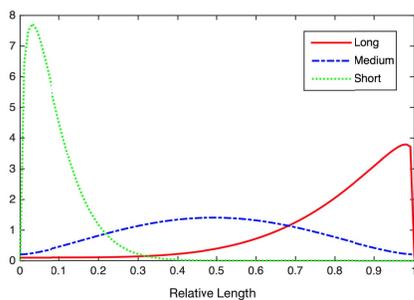


Figure 6. Beta distribution for the relative lengths of long, medium and short strokes.

**Phase 3 — Final design of the handwriting gestures**

Based on the above results, we decided to remove the requirement of one-to-one mapping of gestures to letters, which posed a major challenge for designing handwriting stroke gestures with good memorability and usage efficiency. Instead, we allowed a handwriting stroke gesture to represent more than one letter if necessary. This decision was inspired by an ambiguous keyboard that was also for resolving the limitation of input resources (e.g. a keypad with a small number of buttons). Based on the information gathered during the initial design of the handwriting stroke gestures, we formed the following guidelines for the gesture design.

- (1) Mimic traditional handwriting: 1D Handwriting stroke gestures should be easy to learn. By mimicking traditional handwriting, the obtained strokes should be easy to be remembered.
- (2) Minimize levels of stroke lengths: In order to perform stroke gestures accurately and efficiently.
- (3) Minimize number of sub-strokes per letter: In order to perform stroke gestures efficiently.
- (4) Single stroke input: In order to perform stroke gestures efficiently.

The design goal was to obtain a comprehensive trade-off between speed, accuracy, learnability and memorability to provide immediate usage. One basic principle was to promote simplicity rather than distinction. To achieve this, we first designed a “best” stroke gesture for each individual letter without concerning about ambiguity. We then performed a few adjustments to limit the number of letters in each group to reduce ambiguity and facilitate letter selection within a group. We designed 1D Handwriting stroke gestures to mimic the sub-stroke sequence in two-dimensional handwriting and to be as succinct as possible.



Figure 7. Gesture design for individual characters. The image of Google Glass is placed to the left for reference. Red arrows starting with a circle are the first strokes in traditional handwriting. Letters “e”, “s”, “z” are rotated 90 degrees counterclockwise. Black arrows represent the proposed design for one-dimensional gestures.

Figure 7 illustrates the final stroke design for each letter. The design was based on glyphs of lower-case letters since most of these are performed with a single stroke. In total, we designed 13 stroke gestures, which were mapped to the 26 letters of the alphabet. There were no more than four letters associated with each handwriting stroke gesture. We dealt differently with letters that lacked optimal mappings in a one-dimensional space if mapped directly. For the letters “e”, “z”, and “s”, their glyphs were first rotated 90 degrees counterclockwise and then mapped to a one-dimensional space. Letter “x” has a two-stroke glyph. We thus assigned the gesture of another two sub-stroke letter “n” to it. We also introduce shorthand in the design. In

traditional handwriting, letters “n”, “m”, and “p” start with a backward (toward the body) stroke, while letter “d” ends with a backward stroke. These strokes were eliminated in our design for simplicity and usability.

### THE TECHNIQUE

In this section, we describe the implementation details of the technique used in 1D Handwriting. We first describe the algorithm to recognize gestures, which estimates a posterior probability of an observed gesture given by individual letters. We then present our Bayesian algorithm to predict target words by taking advantage of the language model. We also describe the interaction design, for inputting individual characters and words respectively.

#### Gesture recognition

The purpose is to estimate a posteriori probability of a handwriting stroke input ( $s$ ) given a stroke template ( $t$ ), rather than to discretely classify it into a specific template. Note that according to our design, a stroke template may represent more than one letter. We can then use the estimated result to produce letter-level input and dictionary-based word-level input.

The recognition algorithm has four steps:

##### Step #1: Filtering and segmentation.

We first identify the “turning points” in the stroke, these occur when stroke direction changes. The starting and end points of a stroke are two default turning points. Then we proceed to eliminate consecutive turning points whose distance is less than 20 pixels (1mm) as noise. Finally, segmentation is performed based on the remaining turning points. The derived sequence of sub-strokes is represented as  $s_1 \dots s_n$ .

##### Step #2. Removal of unintentional strokes

In order to deal with unintentional hooks that usually occur at the beginning and end of a handwriting gesture, we remove the first and last sub-strokes that are less than 150 pixels (7.5 mm) in length.

##### Step #3. Normalization

We normalize the length of an individual sub-stroke ( $s_i$ ) by dividing the length of the longest sub-stroke. Note that the length of sub-strokes in the gesture templates are normalized in advance.

##### Step #4. Computation of a posteriori probability

We only compute the template gestures whose number of sub-strokes is equal to the derived gesture. The a posteriori probability of other templates is set to zero. Furthermore, we assume the estimation of individual normalized sub-strokes to be independent from each other. We have

$$P(s|t) = \prod_{i=1}^n P(s_i|t_i) \quad (1)$$

where the probability density function (PDF) of  $P(s_i|t_i)$  is given according to the result of Study 1.

### Predicting target words

A user inputs a sequence of strokes ( $I = s_1 \dots s_n$ ) to input one word ( $w$ ) from a vocabulary ( $V$ ). The word has  $n$  letters, that is,  $w = l_1 \dots l_n$ . The algorithm predicts a number of likely words based on the probability of gesture input and language model. Similar to Goodman et al.’s approach [9], the posterior probability of  $I$  given a word ( $w$ ) can be computed as follows:

$$P(w|I) = P(w, I) * P(I) \quad (2)$$

Because we have no knowledge of  $I$ , we can compute

$$P(w, I) = P(w) * P(I/w) \quad (3)$$

We then assume each letter input is independent from each other. We have

$$P(w, I) = P(w) * \prod_{i=1}^n P(s_i|l_i) \quad (4)$$

That is,

$$P(w, I) = P(w) * \prod_{i=1}^n P(g_i|t(l_i)) \quad (5)$$

where  $t(l_i)$  is the stroke template of letter  $l_i$ .

If no matches are found, the algorithm attempts to make auto-completions by computing the probability of words that have more letters than the input sequence. All predicted words are sorted, according to their probability, and presented to the user in descending order [27].

### Interaction design

Figure 8 shows the interface of the 1D Handwriting system on Google Glass that users can see on the virtual screen. It is divided into three regions. The Text region (the middle row) displays the input text. The Letter region (the top row) displays the recognized letters. The Word region (the bottom row) displays the five most likely words.



Figure 8. Interface design of 1D Handwriting

The handwriting stroke is rendered in real-time on a semi-transparent overlay on top of Text region. The height of the Text region is directly mapped to the entire touchable range of the touchpad. The horizontal direction represents the time line. Therefore, although the input is a one-dimensional stroke gesture, users can still observe a two-dimensional visual feedback of the handwriting stroke on the screen, providing guidance during input.

*Input a word.* Users input words by performing the respective stroke gestures of each letter in sequence. Words that most likely correspond to the current input (at most five) are shown in the Word region. Users can directly select the most likely word out of them by tapping the touchpad with two fingers, or select the other words by performing a two-finger movement to select the desired word, followed by a two-finger tap to confirm. When a word is selected, the system automatically appends a space to the end.

*Input a letter.* The system is set by default to word input mode. In order to input a letter, users have to place their finger on the touchpad for 300ms (empirically determined) after performing a stroke. At that point, the system will switch to letter input mode (with the frame of the Letter region highlighted) allowing users to move their finger on the touchpad to select letters. Lifting the finger will confirm the selection. This design offers a seamless one-finger gesture for inputting letters.

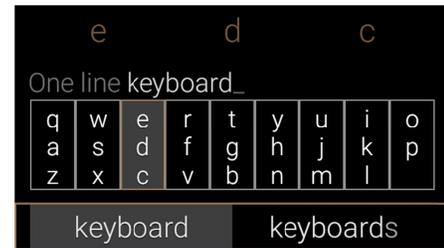
For some letters, the handwriting stroke gesture is likely to end near one of the edges of the touchpad, making further movement in that direction difficult. To deal with this, the system arranges letters in frequency order toward the opposite direction of the last stroke, with the most frequently used letter selected at the endpoint of the stroke.

Deletion of a letter is performed with a swipe-up gesture or by clicking the camera button. According to our observation, most users prefer to use the camera button. Most likely due to the fact that a swipe-up gesture is perpendicular to the main input direction, therefore not easy to be performed on the touchpad.

#### A VARIATION OF THE 1LINE KEYBOARD

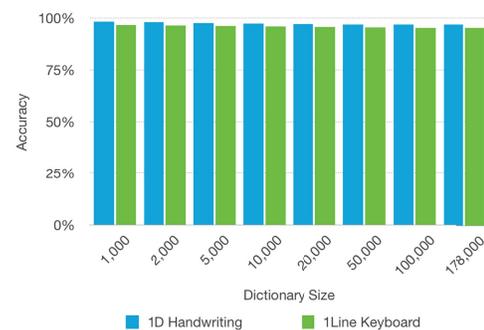
For the sake of comparison, we implemented a selection-based keyboard, referring to 1Line Keyboard design [13], an ambiguous keyboard based on QWERTY layout. As shown in Figure 9, The interface and the interaction are similar to 1D Handwriting except that a user inputs by touch-selection, rather than 1D gesturing: Users manually switch between letter mode and word mode with a swipe-down gesture (in our experiment, the mode was not switched). To input a letter, users swipe one finger to select the ambiguous key, swipe two fingers to select the letter, and then tap two fingers to confirm the selection. To input a word, users swipe one finger to select an ambiguous key, tap one finger to confirm the selection, swipe two fingers to select a word, and then tap two fingers to confirm the selection.

To improve target selection with the touchpad on Google Glass, we implemented cursor acceleration. We used two control/display ratios for slow and fast finger movements respectively. We empirically determined the parameters so that users could traverse the entire keyboard with a single fast stroke or perform accurate selections with slower strokes.



**Figure 9.** A variation of 1Line Keyboard. The currently selected key (“edc”) is highlighted. For consistency purposes, the interface design is similar to 1D Handwriting.

Because both 1Line Keyboard and 1D Handwriting were “ambiguous” for inputting words, we performed a simulation to quantify their ability to specify target words. We used a dictionary (with frequency data) derived from the source code of the Android Keyboard. Figure 10 illustrates the results. Both 1D Handwriting and 1Line Keyboard exhibited high Top-1 accuracy (over 95% for vocabularies containing as many as 178,000 words). In the experiments, we used the top 10,000 words as the dictionary.



**Figure 10.** Accuracy for vocabularies of varying sizes

#### USER STUDY 2: INPUT LETTERS

In order to evaluate the performance of 1D Handwriting for inputting letters, we compared its performance against that of 1Line Keyboard.

#### Apparatus and participants

Sixteen participants (eight males and eight females; aged from 18-30) were recruited from Tsinghua University campus. None of them had previously used a Google Glass before.



**Figure 11.** The experiment setting of User Study 2

Figure 11 illustrates the experiment environment and setting. The system displayed phrases for participants to enter on a laptop PC. Participants sat approximately 0.5m away from the laptop PC. They wore a Google Glass and rested their elbow onto the desk while performing the required inputs. The software on Google Glass logged and synchronized all touch events during the experiment with the laptop PC through Wi-Fi.

### Design and Procedure

We used a within-subject design for this experiment. The only one independent factor was the Technique employed (1D Handwriting and 1Line Keyboard). The tested phrases for each participant was randomly generated from MacKenzie and Soukoreff's phrase set [14]. We used the same phrases for the two techniques per participant, with the order of phrases randomized. Each technique was tested in one session. Sessions consisted of six blocks, with each block containing two phrases for input. Subjects were required to break after finishing a block. The first block was ignored for practice. The presentation order of each technique was counterbalanced.

Before the experiment, we introduced its goals to each participant. We did not inform participants which of the two techniques was ours. Moreover, we allowed subjects to familiarize themselves with the user interface of Google Glass, which took approximately five to ten minutes. Before each technique was presented, participants had a warm-up session with the tested text entry algorithms. During the experiment, subjects were required to input phrases as fast and accurate as possible. Finally, at the end of the experiment, participants filled a NASA-TLX evaluation form and provided feedback on the two techniques. The total experiment lasted for roughly ninety minutes.

The warm-up session of 1D Handwriting was longer than that of 1Line Keyboard. For 1D Handwriting, a sheet containing every letter-gesture pair was first presented to participants. Next, we asked participants to memorize and practice the strokes on Google Glass by correctly performing each gesture five times. Finally, participants were asked to complete a short practice task where all 26 letters appeared twice in a random order. As for the 1Line Keyboard, participants practiced input phrases on Google Glass until they became familiarized with the touch movement control and other input gestures (e.g. selecting/deleting a letter). Warm-up sessions of both techniques lasted for about thirty minutes.

### Results

The main dependent measures were input errors (corrected and uncorrected errors), input speed and input time. To provide more detailed information about 1D Handwriting, we further divided its input time into elapsed time (time between the completion of the previous stroke and the start of the next stroke), gesturing time (time spent on performing the handwriting stroke gesture) and selection

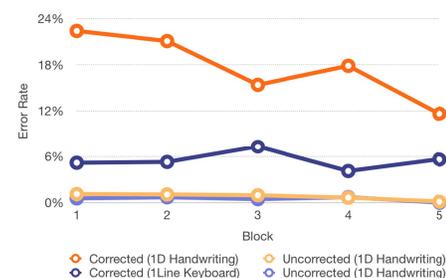
time (time spent on performing the selection). It is important to note that the elapsed time includes the necessary pause between successive handwriting stroke gestures (to complete a stroke gesture and reset the finger location) and the time participants spent to recall the handwriting stroke gestures.

### Corrected and uncorrected errors

Figure 12 illustrates how the corrected and uncorrected error rates migrated with blocks for letter-level input.

There was no significant effect of Technique on uncorrected error ratio ( $F_{1,15}=1.76$ ,  $p=.20$ ). Both techniques exhibited low uncorrected error ratios: 0.78% (SD=1.12%) for 1D Handwriting and 0.48% (SD=0.65%) for 1Line Keyboard. The reason was that participants tended to correct mistakes if they noticed it on the virtual screen.

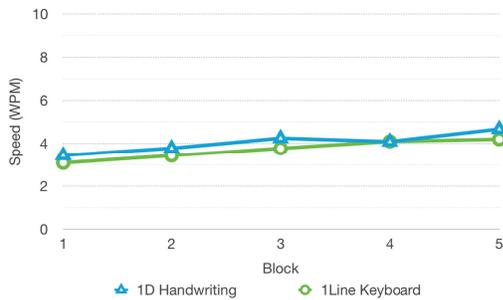
ANOVA showed significant effects of Technique ( $F_{1,15}=68.92$ ,  $p<.0001$ ) and Block ( $F_{4,60}=3.80$ ,  $p<.01$ ) on corrected error ratio. The interaction effect of Technique×Block was also significant ( $F_{4,60}=4.22$ ,  $p<.005$ ). Participants had more corrected errors with 1D Handwriting than with 1Line Keyboard, 17.67% (SD=6.10%) vs. 5.51% (SD=3.69%). However, through practice, participants could perform better (i.e. having less corrected errors) ( $F_{4,60}=4.68$ ,  $p<.005$ ) with 1D Handwriting. In contrast, such a learning effect for 1Line Keyboard was not significant ( $F_{4,60}=1.39$ ,  $p=.29$ ).



**Figure 12. The errors for both techniques with the statistical testing results.**

### Input Speed

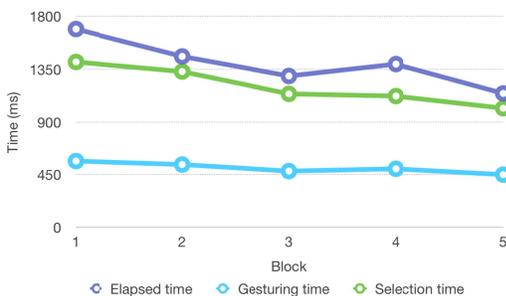
Figure 13 shows the text entry rates for both techniques over the five blocks. The calculation of text entry rate included the time users spent on correcting errors. On average, participants input at 4.04 WPM (SD=0.58) with 1D Handwriting and 3.71 WPM (SD=0.46) with 1Line Keyboard. If we considered only the last block, participants input at 4.67 WPM (SD=0.61) with 1D Handwriting and 4.20 WPM (SD=0.61) with 1Line Keyboard. Hence, 1D Handwriting was 8.9% faster than 1Line Keyboard in average ( $F_{1,15}=5.31$ ,  $p<.05$ ), and 11.2% faster in the last block ( $F_{1,15}=8.70$ ,  $p<.01$ ). Meanwhile, we observed a significant main effect of block on text entry rates for both techniques (1D Handwriting:  $F_{4,60}=14.50$ ,  $p<.0001$ ; 1Line Keyboard:  $F_{4,60}=15.39$ ,  $p<.0001$ ), suggesting the learning effect. There was no interaction effect of Technique×Block ( $F_{4,60}=1.37$ ,  $p=.26$ ) on text entry rates.



**Figure 13. Text entry rates across blocks for 1D Handwriting and 1Line Keyboard. There was a sudden performance drop in the 4<sup>th</sup> block for 1D Handwriting. We interpret it as a noise resulted from uncontrolled factors in the experiment.**

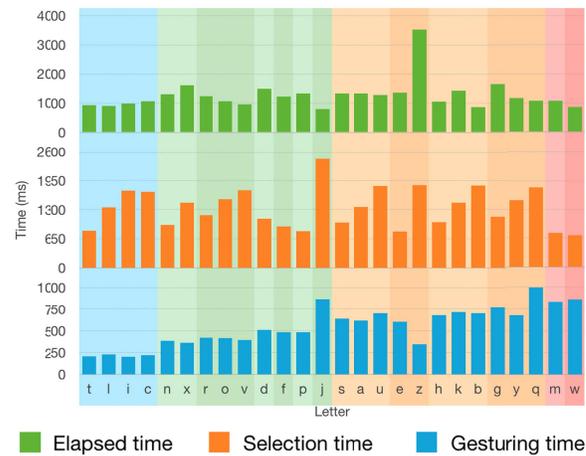
*Input times of 1D Handwriting*

In average, elapsed time was 1397ms (SD=354), gesturing time was 504ms (SD=96), and selection time was 1206ms (SD=151) including 300ms spent on switching modes. Elapsed time, gesturing time and selection time accounted for 45%, 16% and 39% of the input time respectively. That is, the majority of input time was spent on considering the next letter, recalling the handwriting stroke gesture, and selecting letters. Fortunately, with practice, all input times could be improved (Figure 14): RM-ANOVA showed significant effects of Block on elapsed time ( $F_{4,60}=8.73$ ,  $p<.0001$ ), gesturing time ( $F_{4,60}= 5.41$ ,  $p<.001$ ) and selection time ( $F_{4,60}= 11.04$ ,  $p<.0001$ ). A noteworthy aspect is that the learning curve of gesturing time was not as steep as that of elapsed time and selection time.



**Figure 14. Elapsed time, gesturing time and selection time across five blocks for letter-level 1D Handwriting in Study 2.**

Furthermore, analysis of elapsed time, gesturing time and selection time for individual letters was conducted for each letter. From Figure 15, we can see that, gesturing time increased as the number of sub-strokes or the complexity of the structure of sub-strokes increased (with both long and medium length sub-strokes). Selection time was primarily dependent on the number of letters represented by input gestures as well as the order of letters inside a group. There was no obvious pattern for elapsed time. However, we detected that for some letters that were seldom input (e.g. “z”) elapsed time was relatively high, indicating that participants spent more time recalling its handwriting stroke gesture. Interestingly, for the letter “j”, although the elapsed time was short, both gesturing time and selection time were rather large.



**Figure 15. Elapsed time, selection time and gesturing time for different letters. The colors separate letters with respect to the same handwriting stroke for easy comparison.**

*Subjective feedbacks*

Nonparametric Wilcoxon signed-rank tests suggested that compared to 1Line Keyboard, participants felt 1D Handwriting to be more mentally demanding ( $z=-52.5$ ,  $p<.0001$ ). On the other hand, they gave 1D Handwriting a much better evaluation ( $z=-42.0$ ,  $p<.01$ ). Differences on other measures were not statistically significant.

Although at first it seemed difficult to remember and learn to perform the handwriting stroke gestures, participants managed to quickly master the technique in less than ten practice rounds for each letter. In the post-experiment interview, every participant reported 1D Handwriting stroke gestures to be intuitive to recognize and easy to remember. In addition, most participants reported higher levels of eye fatigue with 1Line Keyboard since more focus was required on the keyboard to control cursor movement. In contrast, 1D Handwriting required a lesser degree of focus on the screen while performing stroke gestures, especially when participants were more familiar with the stroke gestures. These findings suggested eyes-free input was possible with 1D Handwriting.

**USER STUDY 3: INPUT WORDS**

The goal of this study was to evaluate and compare 1D Handwriting and 1Line Keyboard for inputting words. The sixteen participants who participated in Study 2 were recruited again for this study. The experiment took place seven to fourteen days after Study 2. We used the same apparatus as in Study 2.

The design and procedure were similar to Study 2, except that participants performed word-level input as opposed to letter-level input. In this experiment, participants were required to enter 18 phrases (divided into nine blocks).

Prior to the experiment, participants were asked to recall the handwriting stroke gestures for all 26 letters of the alphabet, and to write them down on a piece of paper. Results showed that participants could correctly recall 89% of the stroke

gestures. Four letters that participants could not remember well were “x”, “d”, “j” and “t”. Considering the limited training each participant had (no more than 20 times for each letter in average) and the interval of seven to fourteen days, this result positively indicated the good memorability of 1D Handwriting stroke gestures.

Participants familiarized themselves with the techniques in a warm-up session before the task, which lasted around ten to fifteen minutes for each technique. Participants were required to input phrases as fast and accurate as possible. The total experiment lasted for about eighty minutes.

**Results**

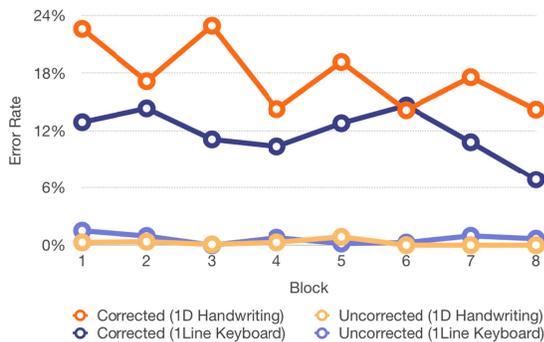
The main dependent measures were input errors (corrected and uncorrected errors) and input speed. All data are with 2 standard deviations. Therefore, no outliers were removed.

*Corrected and uncorrected errors*

Figure 16 illustrates how the corrected and uncorrected error rates migrated with blocks for word-level input.

There was no significant effect of Technique on uncorrected error ratio ( $F_{1,15}=1.21, p=.29$ ). Both techniques exhibited low uncorrected error ratios: 0.25% (SD=0.07%) for 1D Handwriting and 0.07% (SD=1.42%) for 1-Line Keyboard.

We observed a marginal significant effect of Technique on corrected error ratio ( $F_{1,15}=4.51, p=.051$ ). The corrected error ratio was 17.38% (SD=8.67%) for 1D Handwriting and 11.87% (SD=5.81%) for 1Line Keyboard. There was a significant effect of Block on corrected error ratio ( $F_{7,105}=1.15, p=.034$ ), suggesting the learning effect. No significant interaction effect of Technique  $\times$  Block ( $F_{7,105}=0.46, p=.86$ ) was observed.



**Figure 16. The corrected and uncorrected error rates for both techniques across blocks**

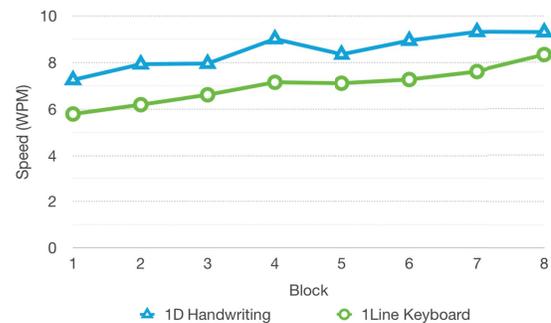
*Input Speed*

Figure 17 shows the text entry rates for both techniques over the eight blocks.

In average, participants input at 8.84 WPM (SD=2.79) with 1D Handwriting and 7.14 WPM (SD=1.21) with 1Line Keyboard. That is, 1D Handwriting was 23.8% faster than 1Line Keyboard in average ( $F_{1,15}= 6.92, p<.05$ ). If only the last two blocks were considered, text entry rates raised to 9.72 WPM (SD=2.67) for 1D Handwriting and 8.10 WPM

(SD=1.34) for 1Line Keyboard. That is, 1D Handwriting was 20.0% faster than 1Line Keyboard in the last two blocks ( $F_{1,15}=5.86, p<.05$ ).

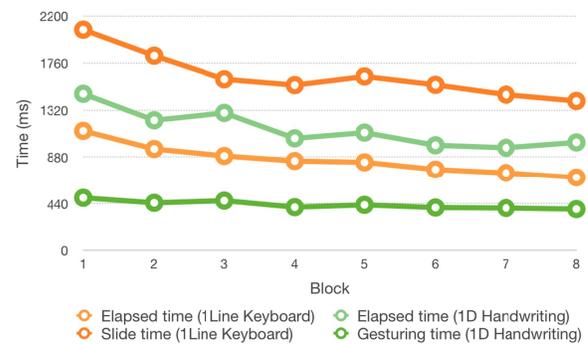
ANOVA also showed a significant main effect of Block on text entry rates ( $F_{7,105}=11.43, p<.0001$ ), suggesting the learning effect. There was no significant interaction effect of Technique $\times$ Block on text entry rates ( $F_{7,105}=1.25, p=.28$ ).



**Figure 17. The learning curve of both techniques.**

*Input times*

On average, the elapse time of 1D Handwriting was 289 ms (SD=66) longer than that of 1Line Keyboard, which could be interpreted as the recall time for gestures. In addition, when compared to the slide time (1,593 ms, SD=132) of 1Line Keyboard, the gesturing time (427 ms, SD=38) for 1D Handwriting was much smaller. Similar to the results of Study 2, we observed significant learning effects for both techniques ( $p<.05$ ), as shown in Figure 18.



**Figure 18. Slide time, gesturing time and elapsed time and across eight blocks for both techniques in Study 3.**

*Auto-completion*

Auto-completion accounted for 6.51% of input letters for 1D Handwriting and 38.10% for 1Line Keyboard. The difference was statistically significant ( $F_{1,15}=191.49, p<.0001$ ). In other words, if we did not consider letters input via auto-completion, participants would have input phrases at 8.30 WPM (SD=2.70) for 1D Handwriting, and 5.17 WPM (SD= 0.86) for 1-Line keyboard over all blocks. The reason why participants took more advantage of auto-completion for 1Line Keyboard was as follows: With 1D Handwriting, as participants performed the handwriting stroke gestures, they tended to look at the test phrases on the laptop PC screen instead of the Google Glass display.

They only turned to the virtual screen whenever they needed to select words. In contrast, with 1Line Keyboard, participants constantly looked at the virtual screen to control cursor movement. Therefore, it was more likely for participants to notice the auto-completion suggestions. In the post-experiment interview, we confirmed this difference with participants. We think in actual real-life scenarios, participants may be more likely to take full advantage of the auto-completion function with 1D Handwriting by having their eyes rested on the virtual screen.

#### USER STUDY 4: EXTENSIVE TRAINING

The goal of this study was to estimate how fast a user could be following extensive training. To achieve this, we asked five participants who participated the Study 3 to repeatedly input a single phrase (randomly selected from MacKenzie and Soukoreff's phrase set [14]) for twelve times. The first two trials were ignored as practice. By focusing on a small number of words, the training time could be greatly reduced [6]. However, it is important to note that given the practice was based on a single phrase, the obtained speed could only served as a rough estimation. In this study, we switched off auto-completion so that participants had to input each letter of a word with handwriting stroke gestures.

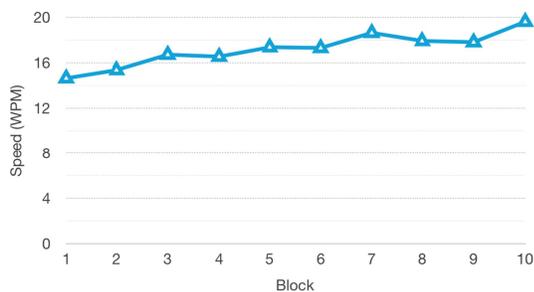


Figure 19. The learning curve of 1D Handwriting in Study 4.

Results showed the mean text entry rate could reach 19.61 WPM (SD=3.50) for the last phrase. The fastest participant achieved 24 WPM, while the slowest participant achieved 16 WPM. As Figure 19 shows, there was a significant learning effect of Block ( $F_{9,63}=6.16, p<.0001$ ) on text entry rate. In average, participants entered 33.95% faster in the last block than in the first one.

According to participants' subjective feedback and our own observations, a major factor for prohibiting high text entry rates was the limited input capacity of Google Glass. When performing fast gestures, the finger often went off or overshoot the touchpad. In such cases, participants had to slow down finger movements to produce correct inputs. Moreover, we confirmed with participants with regard to the design of experiment: Participants thought the input speed and experience obtained in the experiment could reasonably reflect their proficient input speed if they had used 1D Handwriting for a longer period.

#### LIMITATIONS AND FUTURE WORK

This research has the following limitations, which also point to the direction of future works.

First, our current 1D handwriting gestures were designed for lower-case English letters. It is important to support more character sets, such as numbers, punctuations and non-Latin alphabets. One requirement is that there should be an easy-to-recognize mapping between 2D strokes and 1D strokes; otherwise, the 1D strokes will be difficult to remember. In addition, supporting more characters means overloading a single stroke with more characters. One can address this with mode selection to specify different character sets. For example, on Google Glass, we can use the "flip up" gesture to switch between upper case and lower case, or even between numbers and punctuations

Second, regarding learnability, our research showed that participants could learn the 1D strokes with a few practices, and recall 89% of the strokes after one week. However, there is still space for improvement. For example, we can employ a well-designed learning interface (e.g. OctoPocus [1]) to facilitate gesture learning. Moreover, it is interesting to test the performance after long-term and extensive training. Now, in all three experiments, the input speed of 1D Handwriting did not converge due to the limited learning time.

Third, in this research, we confirmed the feasibility of 1D handwriting. However, the current interaction design may be not optimal for Google Glass: We did not take full advantage of the vertical input and other input sensors (such as the three-point touch and even the embedded motion sensors) for input. It is also interesting to test 1D handwriting technique on other 1D interfaces such as smartphones with a curved-edged screen (e.g. Galaxy S6 edge), smart wristbands, or even 2D touchscreens.

Finally, in current research, we focus on 1D handwriting for inputting letters. It is interesting to explore the possibility of 1D handwriting for word level input, i.e. 1D gesture keyboard. This might further improve the text entry speed.

#### CONCLUSION

We explored the possibility of text entry on an emerging form of constrained interface: the one-dimensional interface. The basic idea was to project two-dimensional handwriting strokes into a one-dimensional space. We analyzed the usability issues of designing unistroke 1D Handwriting gestures. Subsequently, we designed, implemented and evaluated 1D Handwriting with Google Glass. In user studies, both objective and subjective results suggested 1D Handwriting stroke gestures were easy to remember and efficient to use. Our research verified the feasibility of 1D Handwriting.

#### ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China under Grant No. 61303076 and No. 61272230, Tsinghua University Research Funding No. 20151080408 and No. 20151300771.

## REFERENCES

1. Olivier Bau and Wendy E. Mackay. 2008. OctoPocus: a dynamic guide for learning gesture-based command sets. In *Proceedings of the 21st annual ACM symposium on User interface software and technology* (UIST '08), 37-46.  
<http://dx.doi.org/10.1145/1449715.1449724>
2. Conrad H. Blickenstorfer. 1995. Graffiti: Wow! *Pen Computing Magazine*, pp. 30-31.
3. Steven J. Castellucci and I. Scott MacKenzie. 2013. Gestural text entry using Huffman codes. In *Proceedings of the International Conference on Multimedia and Human-Computer Interaction* (MHCI '13), 119.1-119.8.  
<http://www.yorku.ca/mack/mhci2013e.html>
4. Steven J. Castellucci and I. Scott MacKenzie. 2008. Unigest: text entry using three degrees of motion. In *CHI '08 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '08), 3549-3554.  
<http://dx.doi.org/10.1145/1358628.1358889>
5. Steven J. Castellucci and I. Scott MacKenzie. 2008. Graffiti vs. unistrokes: an empirical comparison. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '08), 305-308.  
<http://dx.doi.org/10.1145/1357054.1357106>
6. Xiang 'Anthony' Chen, Tovi Grossman, and George Fitzmaurice. 2014. Swipeboard: a text entry technique for ultra-small interfaces that supports novice to expert transitions. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (UIST '14), 615-620.  
<http://dx.doi.org/10.1145/2642918.2647354>
7. Torsten Felzer and Rainer Nordmann. 2006. Alternative text entry using different input methods. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility* (Assets '06), 10-17.  
<http://dx.doi.org/10.1145/1168987.1168991>
8. David Goldberg and Cate Richardson. 1993. Touch-typing with a stylus. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (CHI '93), 80-87.  
<http://dx.doi.org/10.1145/169059.169093>
9. Joshua Goodman, Gina Venolia, Keith Steury, and Chauncey Parker. 2002. Language modeling for soft keyboards. In *Proceedings of the 7th international conference on Intelligent user interfaces* (IUI '02), 194-195.  
<http://dx.doi.org/10.1145/502716.502753>
10. Zhenyu Gu , Xinya Xu, Chen Chu, Yuchen Zhang. 2015. To Write not Select, a New Text Entry Method Using Joystick. *Human-Computer Interaction: Interaction Technologies*, 35-43.
11. Eleanor Jones, Jason Alexander, Andreas Andreou, Pourang Irani, and Sriram Subramanian. 2010. GesText: accelerometer-based gestural text-entry systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '10), 2173-2182.  
<http://dx.doi.org/10.1145/1753326.1753655>
12. Per-Ola Kristensson and Shumin Zhai. 2004. SHARK<sup>2</sup>: 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 '04), 43-52.  
<http://dx.doi.org/10.1145/1029632.1029640>
13. Frank Chun Yat Li, Richard T. Guy, Koji Yatani, and Khai N. Truong. 2011. The 1line keyboard: a QWERTY layout in a single line. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (UIST '11), 461-470.  
<http://dx.doi.org/10.1145/2047196.2047257>
14. I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '03), 754-755.  
<http://dx.doi.org/10.1145/765891.765971>
15. I. Scott MacKenzie and Shawn X. Zhang. 1997. The immediate usability of Graffiti. *Graphics Interface 97*: 129-137.
16. I. Scott MacKenzie, R. William Soukoreff, and Joanna Helga. 2011. 1 thumb, 4 buttons, 20 words per minute: design and evaluation of H4-writer. In *Proceedings of the 24th annual ACM symposium on User interface software and technology* (UIST '11), 471-480.  
<http://dx.doi.org/10.1145/2047196.2047258>
17. Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. 2014. Vulture: a mid-air word-gesture keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14), 1073-1082.  
<http://dx.doi.org/10.1145/2556288.2556964>
18. Franck Poirier and Mohammed Belatar. 2007. UniGlyph: only one keystroke per character on a 4-button minimal keypad for key-based text entry. *HCI International 2007*, 479-483.
19. Shuo Qiu, Kyle Rego, Lei Zhang, Feifei Zhong, and Michael Zhong. MotionInput Gestural Text Entry in the Air.  
<http://www.columbia.edu/~sq2144/MotionInput.pdf>

20. Garth Shoemaker, Leah Findlater, Jessica Q. Dawson, and Kellogg S. Booth. 2009. Mid-air text input techniques for very large wall displays. In *Proceedings of Graphics Interface 2009* (GI '09), 231-238.
21. The Minuum Keyboard. <http://minuum.com/>
22. William S. Walmsley, W. Xavier Snelgrove, and Khai N. Truong. 2014. Disambiguation of imprecise input with one-dimensional rotational text entry. *ACM Trans. Comput.-Hum. Interact.* 21, 1:4 <http://dx.doi.org/10.1145/2542544>
23. Wiimote, [https://en.wikipedia.org/wiki/Wii\\_Remote](https://en.wikipedia.org/wiki/Wii_Remote)
24. Jacob Wobbrock and Brad Myers. 2006. Trackball text entry for people with motor impairments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '06), 479-488. <http://dx.doi.org/10.1145/1124772.1124845>
25. Jacob O. Wobbrock, Htet Htet Aung, Brandon Rothrock, and Brad A. Myers. 2005. Maximizing the guessability of symbolic input. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '05), 1869-1872. <http://dx.doi.org/10.1145/1056808.1057043>
26. Jacob O. Wobbrock, Duen Horng Chau, and Brad A. Myers. 2007. An alternative to push, press, and tap-tap-tap: gesturing on an isometric joystick for mobile phone text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '07), 667-676. <http://dx.doi.org/10.1145/1240624.1240728>
27. Jacob O. Wobbrock, Brad A. Myers, and Duen Horng Chau. 2006. In-stroke word completion. In *Proceedings of the 19th annual ACM symposium on User interface software and technology* (UIST '06), 333-336. <http://dx.doi.org/10.1145/1166253.1166305>
28. Jacob O. Wobbrock, Brad A. Myers, and John A. Kembel. 2003. EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the 16th annual ACM symposium on User interface software and technology* (UIST '03), 61-70. <http://dx.doi.org/10.1145/964696.964703>
29. Jacob Wobbrock, Brad Myers, and Brandon Rothrock. 2006. Few-key text entry revisited: mnemonic gestures on four keys. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '06), 489-492. <http://dx.doi.org/10.1145/1124772.1124846>
30. Jacob O. Wobbrock, Brad A. Myers, Htet Htet Aung, and Edmund F. LoPresti. 2003. Text entry from power wheelchairs: edgewrite for joysticks and touchpads. In *Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility* (Assets '04), 110-117. <http://dx.doi.org/10.1145/1028630.1028650>
31. Shumin Zhai and Per-Ola Kristensson. 2003. Shorthand writing on stylus keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '03), 97-104. <http://dx.doi.org/10.1145/642611.642630>