Tap, Dwell or Gesture?: Exploring Head-Based Text Entry Techniques for HMDs

Chun Yu[1] **Yizheng Gu**[1] **Zhican Yang**[1] **Xin Yi**[1] **Hengliang Luo**[2] **Yuanchun Shi**[1]
[1]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
[2]Samsung Beijing R&D Center, Beijing, 100028, China
{chunyu, shiyc}@tsinghua.edu.cn, {guyz13, yang-zc13, yix15}@mails.tsinghua.edu.cn, hl.luo@samsung.com

ABSTRACT
Despite the increasing popularity of head mounted displays (HMDs), development of efficient text entry methods on these devices has remained under explored. In this paper, we investigate the feasibility of head-based text entry for HMDs, by which, the user controls a pointer on a virtual keyboard using head rotation. Specifically, we investigate three techniques: *TapType*, *DwellType*, and *GestureType*. Users of *TapType* select a letter by pointing to it and tapping a button. Users of *DwellType* select a letter by pointing to it and dwelling over it for a period of time. Users of *GestureType* perform word-level input using a gesture typing style. Two lab studies were conducted. In the first study, users typed 10.59 WPM, 15.58 WPM, and 19.04 WPM with *DwellType*, *TapType*, and *GestureType*, respectively. Users subjectively felt that all three of the techniques were easy to learn and considered the induced fatigue to be acceptable. In the second study, we further investigated *GestureType*. We improved its gesture-word recognition algorithm by incorporating the head movement pattern obtained from the first study. This resulted in users reaching 24.73 WPM after 60 minutes of training. Based on these results, we argue that head-based text entry is feasible and practical on HMDs, and deserves more attention.

Author Keywords
HMD; Head-based text entry; Dwelling; Gesture keyboard.

ACM Classification Keywords
H.5.2. [Information interfaces and presentation]: User Interfaces-Input devices and strategies.

INTRODUCTION
Head-mounted displays (HMDs) are expected to be the main platform for VR/AR applications, such as VR movies, virtual shopping, chatting and so on. However, development of efficient text entry methods on these devices has remained under explored. Although touch-based [12,23,34] and mid-air [5,24,33] text entry techniques have been proposed, they are either not efficient enough or require expensive peripheral devices (e.g. a camera or a glove).

In this paper, we investigate the feasibility of head-based input techniques for text entry on HMDs, by which, the user controls a pointer on a virtual keyboard using head rotation. To our knowledge, studies of head-based text entry on HMDs do not exist in literature. We expect that HMDs will largely favor head-based interaction, such as text entry. Specifically, we propose and explore three HMD text entry techniques:

- *TapType*: Resembling tap-typing on smart phones, users move a pointer with head rotation and select a letter by tapping a button.
- *DwellType*: A hands-free input method, by which, users select a letter by dwelling over it for a period of time.
- *GestureType*: Enables users to perform word-level input using a gesture based typing style.

We conducted two lab studies. The first study evaluated and compared the performance of the three proposed techniques. Results showed that *DwellType*, *TapType* and *GestureType* yielded text entry rates of 10.59 WPM, 15.58 WPM and 19.04 WPM respectively, with an uncorrected error rate below 0.5%. Meanwhile, users found all three head-based text entry techniques were easy to learn and considered the induced fatigue to be acceptable.

In the second study, we further investigated *GestureType*. We improved the gesture-word recognition algorithm by incorporating head movement patterns obtained in the first study. This resulted in around an 8% accuracy improvement for predicting the most probable candidate. With the new algorithm, users reached 24.73 WPM on average after eight practice sessions of ten sentences, with the best performer achieving 39 WPM. Meanwhile, we observed a significant learning effect in that users improved their input speed by leveraging the correction ability of the gesture-word recognition algorithm.

CHI 2017, May 06-11, 2017, Denver, CO, USA
 ISBN 978-1-4503-4655-9/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3025453.3025964

Therefore, we argue that head-based text entry is both feasible and practical on HMDs. We conclude this research with a discussion of the limitations and future work.

RELATED WORK

We reviewed text entry techniques on HMDs, gaze- and head-based text entry, and gesture keyboards.

Text Input on HMDs

Touch input has been explored for enabling text entry on HMDs. Using a handheld touchpad to move a cursor on a virtual keyboard, users could type 5.01 words per minute [23]. To enable text entry on the Google Glass, Swipezone (8.73 WPM) [12] divides the side touchpad into three zones. Users tap or swipe in a zone to specify a subgroup of letters, and then perform a second touch to select the desired letter. 1D Handwriting (9.72 WPM) [34] employs a different method, in which users input text by handwriting one-dimensional letters on the Google Glass' touchpad.

Mid-air input is another potential means for text entry on HMDs. With air writing [2, 29], users could write 11 words per minute in 3D space [24]. With chording (e.g. Pinch Keyboard [5]), where individual characters are mapped to multi-finger gestures, expert users could type 12-15 words per minute [5]. Sridhar et al. [30] optimized a set of multi-finger gestures by considering both input performance and learnability. The text entry rate was able to reach 22 WPM after repetitive practicing. Moreover, indirect typing with a single-finger (13.2 WPM) [21] and ten-finger touch-typing in the air 29.2 WPM [33] have also been investigated. However, a drawback of mid-air techniques is they usually require additional hardware, such as cameras or gloves.

Gaze- and Head-based Text Entry

A substantial body of work has been conducted on gaze typing. Many are dwell-based: A user inputs a letter by gazing at it for a specific time period (dwell time). As reviewed in [19], most techniques used a constant dwell time between 450ms and 1000ms, and yielded text entry rate from 5 to 10 WPM. To investigate the lower bound of acceptable dwell time, Majaranta et al. [20] conducted an experiment in which users progressively decreased the dwell time to improve performance. After ten 15-minute sessions, the mean dwell time decreased from 876ms to 282ms, while text entry rate increased from 6.9 to 19.9 WPM.

Dwell-free techniques have also been explored. Kristensson et al. [14] showed that the theoretical performance of dwell-free gaze type could reach 46 WPM. That is twice the input speed of the fastest dwell-based techniques [20]. However, it is challenging to process continuous gaze gestures into words due to various sources of noises [26]. EyeK [28] replaces dwell operation with moving the eye pointer through the key in an inside-outside-inside fashion (6.03 WPM). EyeSwipe (11.7 WPM) [16] requires users to accurately select the first and the last characters using reverse crossing, and glance through the vicinity of the middle characters in sequence. By doing so, a large number of unlikely candidate words can be pruned before applying pattern matching [15]. Filteryedping [26] recognizes the intended word by filtering extra letters from the sequence of letters gazed at by the user. Users typed 15.95 WPM after 100-minutes of practicing.

By comparison, there are fewer studies on head-based typing. Gizatdinova et al. [9] asked users to point at the keys of a virtual keyboard with gaze or head, and confirm the selection with a spacebar. They found users could type 10.98 WPM with gaze, and 4.42 WPM with head. With a 500ms dwell-based design, Hansen et al. [13] reported a head typing speed of 6.10 WPM on a dynamic keyboard. However, to our knowledge, no research has investigated the performance of head typing on HMDs, where the display is tightly attached to users' head. We think that head-based input should be favored due to this characteristic.

In this research, we identify three advantages of head typing over gaze typing on HMDs: First, head tracking is low-cost (e.g. with inertial sensor) and already built-in on many commercial HMDs (e.g. Oculus Rift and Samsung VR Gear). Second, previous research has shown that head input was more accurate than gaze [8, 13]. This would ease the design of the recognition algorithm and help to improve performance. Third, while typing with head, users can still perform visual search by moving their eyes when uncertain about key locations.

Word-level Gesture Keyboard

Word-level gesture keyboards (WGKs) were first introduced by Zhai and Kristensson [15,37] to speed up text input using a stylus. On WGKs, users write a word with a continuous gesture path traversing the letters on a virtual keyboard in sequence. WGKs recognize the intended word by matching the gesture to templates of all possible words in a dictionary. Today, WGKs have been widely deployed on smart phones [38]. In a recent study, WGK yielded 30 WPM in lab and 39 WPM in the wild [27].

Researchers have applied WGKs to other input platforms. Bimanual Gesture Keyboard [4] allows users to type gestures with two thumbs on the split keyboard on tablets (39 WPM for experts). WatchWriter [11] demonstrates WGKs work well on smart watch devices with a much smaller touchscreen (24.3 WPM). Vulture [22] adapts WGKs to mid-air, where users remotely perform gestures on a virtual keyboard (20.6 WPM for novices). However, WGKs have not been applied to head-based text entry.

HEAD-BASED TEXT ENTRY ON HMDS

In this section, we describe the design and implementation of the three proposed head-based text entry techniques (*TapType*, *DwellType* and *GestureType*) on HMDs. All three techniques share the same software interface as shown in Figure 1. We rendered the virtual keyboard ten meters far away from the user to avoid parallax. To determine the

appropriate keyboard size, we ran a pilot study with six participants. We test three keyboard width: 4, 6 and 8 meters. 5 participants preferred the 6-meter width keyboard for its comfortableness of viewing and head moving. This yielded a field-of-vision (FOV) of 33.4 degree. Finally, we set the control/display ratio to be 1:1 to ensure a direct sense of head-based pointing [6].

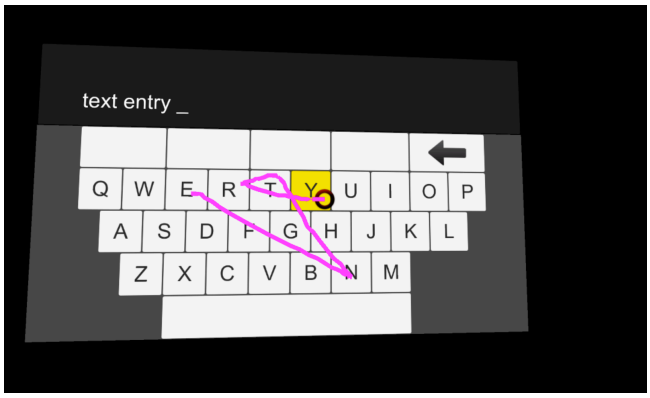

Figure 1: The virtual keyboard interface for *GestureType*. Interfaces for *TapType* and *DwellType* are similar.

The basic interaction concepts of the three techniques are straightforward and have been explained. We now provide more details on the design and implementation.

For *DwellType*, we tested a range of time period, and found 400ms to be appropriate, under which users easily committed unintentional selections. In this research, we did not progressively reduce dwell time as Majaranta et al. did in [20]. Therefore, the tested text entry rate should not represent the optimal performance of dwell-based typing. Moreover, another error-prone action was dwelling on the same key for too long before moving the cursor (possibly while searching the next key), resulting in an error of double selection. To deal with this, we set the dwell time to be 800ms after a key was activated and the cursor did not move off of it.

For *DwellType* and *TapType*, the system displays the literal letters input by users in the text field, which we refer to as the default word in this paper. Users confirm the default word by clicking on the spacebar. Meanwhile, the system performs error correction (we will describe it later), and displays another four likely words in the candidate region, with the best candidate displayed in the first. Users select a suggested word by directly clicking on it. If a word has just been confirmed (or selected), clicking on "Backspace" will delete the word; Otherwise, the letter that has just been entered will be deleted.

For *GestureType*, users press down the button to indicate the start of the gesture, and release it to indicate the end of the gesture. The system applies pattern matching to recognize the input gestures and convert them into words [15]. The predicted word with the highest probability (the default word) is directly displayed in the text field, with four other possible words in the candidate region. Users confirm the default word by directly gesturing the next word, or select another candidate by clicking on it. The system automatically appends a space after a word has been input. Clicking on Backspace deletes the word that has just been entered.

Statistical Decoding Algorithm

For *DwellType* and *TapType*, we apply the statistical decoding method [10] to handle the noise of the input. This method is widely used in modern software keyboards to deal with the "*fat finger*" problem [32]. The basic idea is to compute a word with a maximum posterior probability of the observed input.

Let $\mathrm{I} = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^{n}$ denote the sequence of coordinates of endpoints the user inputs on the keyboard. According to the Bayes rule, the posterior probability of a candidate word W given input I is computed as

$$\mathrm{P(W|I)} = \mathrm{P(I|W)P(W)/P(I)}$$

where P(W) is the prior probability of W being needed by the user, which can be determined as the frequency of words in a corpus; P(I) is constant for all candidates, which can be ignored here. Further, we assume individual key acquisition actions are independent of each other [10]. Thus, we have

$$\mathrm{P(I|W)} = \prod_{i=1}^{n} P(x_i, y_i|c_i)$$

where $P(x_i, y_i|c_i)$ is the distribution of endpoints for c_i (the i^{th} letter in W). In literature, it is often assumed to be a 2D Gaussian distribution. Thus, to determine the parameters, we collected 500 tapping points from real users. We calculated the distribution to have a zero mean and a standard deviation of 0.75 and 0.38 on X-axis and Y-axis, respectively. The unit of measurement is one key-width.

Word-Gesture Recognition

We implemented the gesture-word recognition algorithm by referring to SHARK^2 [15] and Vulture [22]. According to our observation, head-based pointing was more accurate than finger input. Hence, we only utilized the location channel [15]. We now briefly describe the algorithm.

WGKs decode gestures input into words that are predefined in a dictionary. Before processing, each dictionary word is transformed into a template: the lines connecting the key centroid of sequential letters in the word. To handle a gesture, the algorithm first prunes words in the dictionary whose start/end location is farther than 1 key-width from the start/end of the gesture [22]. Then, the algorithm computes the "distance" between a candidate word and the input gesture, which is defined as the sum of the Euler distance between each pair of corresponding points. In our implementation, both the gesture and the template of a word are sampled into $N = 50$ equidistant points [22].

$$D = \sum_{i=1}^{N} \|u_i - t_i\|$$

where ***u*** is the unknown shape that is being compared to the template word ***t***. The word with the minimal distance is recognized as the best match.

For all three of the techniques, we used a 10K lexicon, which contained the most probable words derived from the American National Corpus [1]. The lexicon also contains the frequency of words. Previous research showed 10K words could cover 90% of language use in daily life [25].

STUDY 1: EVALUATING THREE TECHNIQUES

The goal was to evaluate the text entry rate of the three head-based text entry techniques. We also investigated usability issues such as fatigue and learnability.

Participants

Eighteen participants (12 males and 6 females; aged from 18 to 27, M = 21.56) were recruited from the campus. All participants were familiar with the QWERTY keyboard (M = 4, from 1 – No skill to 5 – Expert) according to self-report. Thirteen participants had previous experience with HMDs before.

Apparatus

Figure 2: The experimental setting in Study 1. A participant entered text on an HMD device with head rotation.

The experiment was conducted on Samsung's Gear VR with an S6 Edge Plus smartphone, which afforded a 96-degree field of vision in the HMD. We employed a Bluetooth gamepad to provide the button input. The system leveraged built-in sensors on the phone to capture head rotation. We implemented the three head-based text entry techniques and the experimental system in Unity 3D. Our software system logged gesture data (including timestamp for each point), and interaction operations such as selection and deletion of words.

Design

Since all of the techniques used head rotation for input, we employed a between-subject design to avoid potential cross-learning effect between the three techniques. Each technique was tested on 6 participants (4 males and 2 females). Each participant transcribed 48 phrases in 6 sessions, with each session containing 8 phrases. The 48 phrases were randomly sampled from the MacKenzie phrase set [18]. The two independent factors were Technique (*DwellType*, *TapType* and *GestureType*) and Session.

Procedure

Before testing, we described to participants the goal of the experiment and the input method of the technique that was to be tested. We told participants that error correction was supported. Participants then familiarized themselves with the technique. This warm-up phase usually took no more than three minutes. We then instructed them to enter text "as quickly and accurately as possible". Between sessions, participants took a 1-minute break. After the experiment, we interviewed participants, and asked them to comment on the technique that was tested. This experiment took about 50 minutes.

Result

Text entry rate is measured in Words Per Minute (WPM), with this formula

$$WPM = \frac{|S|}{T} \times 60 \times \frac{1}{5}$$

where $|S|$ is the length of the transcribed string, and T is time in seconds.

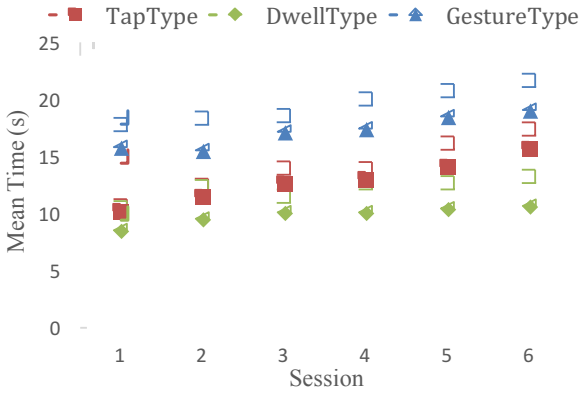

Figure 3: Text-entry rates of three techniques over sessions in Study 1. Error bars indicate standard deviation.

Fig. 3 shows the mean text entry rate over sessions for each technique. Mixed ANOVA showed significant effects of Technique ($F_{2,15} = 19.7$, $p < .0001$) and Session ($F_{5,75} = 65.5$, $p<.0001$) on text entry rate. Among the three tested techniques, *GestureType* was the fastest. Participants typed 15.75 WPM (SD = 2.62) in the first session, and achieved 19.04 WPM (SD = 3.26) in the final session. *TapType* was the second fastest. Participants typed 10.14 WPM (SD = 1.13) at the beginning, and reached 15.58 WPM (SD = 2.42) in the final session. *DwellType* was the slowest. Participants typed 8.48 WPM (SD = 1.16) in the first session and achieved 10.59 WPM (SD = 1.07) in the final session.

Word-Level Uncorrected and Corrected Error Rate

The corrected error rate and uncorrected error rate of all three techniques were found to be low. For *TapType*, *DwellType* and *GestureType*, the mean uncorrected error

rates were 0.57%, 2.46%, and 1.13% respectively, and the mean corrected error rates were 1.45%, 1.23%, and 3.08% respectively. ANOVA showed no significant effects of Technique on corrected error rate or uncorrected error rate.

Interaction Statistics

We examined details on how users interacted with the three techniques. We investigated three word-level actions (Table 1): Default, Select and Delete. Default means that the user confirmed the default word in the text field. For *TapType* and *DwellType*, it was the literal letters that had been input. For *GestureType*, it was the Top-1 word that was predicted. *Select* means the user selected a predicted word from the candidate region other than the default word. *Delete* means the user deleted a word that had just been input.

Action	Y/N	*TapType*	*DwellType*	*GestureType*
Default	**Y**	381	1425	1318
	N	3	37	14
Select	**Y**	1193	82	254
	N	6	1	4
Delete	**-**	23	19	49
Total	**-**	1606	1564	1639

Table 1: The number of word-level interactions done with the three head-based techniques. Y/N indicates whether the input word was correct or not.

For *DwellType*, 93.48% of the actions were Default meaning that the participant entered every key of the intended word. In 5.31% of the actions, the users leveraged error correction meaning that the user dwelled on the wrong key but the system successful corrected the input. The users deleted the word in only 1.21% of the actions. These results indicate that users were quite accurate at pointing in *DwellType*. This is probably because by dwelling, users looked at the letters when confirming selection. Due to this, it was unlikely that a wrong letter was input.

For *TapType*, the results were quite different from *DwellType*. Only 23.91% of actions were Default meaning that the participant clicked inside every key of the intended word. In 74.65% of the actions, users leveraged error correction to input and selected a word in the candidate region. The users deleted the word in only 1.43% of the actions. These results showed that with *TapType*, participants relied more on the correction ability of the algorithm to input, rather than selecting keys accurately.

For *GestureType*, the most frequent action was Default (81.27%) meaning that the user wrote a word gesture and the system decoded it as the correct intended word. In 15.74% of the actions, the user selected a word other than the default one. In only 2.99% of the actions, users deleted the word they had written in order to input again. These results showed the gesture-word recognition algorithm performed well. For 97% of the actions, it successfully interpreted input either as the best match or by suggesting the intended in the candidate region.

Subjective Interview

Fig. 4 shows the subjective feedback of participants. Since this study followed a between-subject design, participants gave these scores independently, without knowing about other techniques.

We observe a significant effect of Technique on Perceived Speed ($F_{2, 15}$=10.5, $p < .01$). Post hoc pairwise comparisons with Bonferroni correction showed the difference between *DwellType* and the other two techniques to be significant, while the difference between *TapType* and *GestureType* was not significant. Participants found all three techniques were easy to learn (M = 4.39, SD = 0.70), and subjectively perceived *TapType* and *GestureType* to be fast (M = 3.83, SD = 0.75; M = 4.33, SD = 0.82), and *DwellType* to be slower (M = 2.5, SD = 0.55). It was also good to find that participants considered the induced fatigue of all three techniques to be acceptable (M = 2.61, SD = 1.03). Overall, participants liked *TapType* and *GestureType* more than *DwellType*.

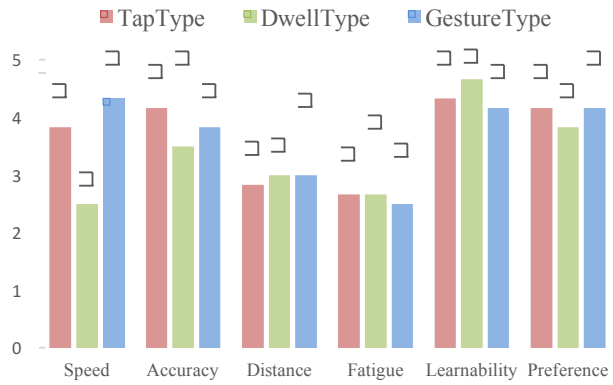

Figure 4: Subjective feedback of participants over three techniques (from 1 - not good to 5 - good) in Study 1. Error bars indicate standard deviation.

IMPROVING Word-Gesture Recognition

	σ_x	σ_y	a	b
Start	0.2455	0.1513	1.27	0.79
End	0.2694	0.1859	1.20	0.83
Middle	0.3856	0.2213	1.32	0.76

Table 2. The standard deviation of distance of the start, the end and the middle points of gestures in Study 1. a and b are the parameters to define the Mahalanobis distance.

In previous research, word-gesture recognition algorithms were researched for finger/pen input. In our research, the human head was used as the pointing device. Therefore, there was the question of whether head input had any distinct patterns that would affect the design of the algorithm. To investigate this issue, we used the gesture data from the first study to analyze the distance between

gesture and the word template at the start, at the end, as well as at the middle points. We sampled 50 equidistant points on each gesture; middle points were all 48 points except start point and end point.

Results are summarized in Table 2, from which, we had two major findings. First, users performed more accurately in Y-axis than in X-axis. The mean distance on X-axis was 1.62, 1.45 and 1.74 times of that on Y-axis at the start, the end and middle points of gestures respectively. In addition, we found head rotation speed on X axis (5.32 key/s) was 3.41 times of that on Y axis (1.56 key/s). This finding inspired us to place different weight on X-axis and Y-axis for interpreting the gesture. To achieve this, we used the Mahalanobis distance instead of the Euler distance which had been used in our first study as well as in all previous research [15, 22]. In our research, the Mahalanobis distance was defined as

$$D_{\mathrm{M}} = \sqrt{\frac{\Delta x^2}{a^2} + \frac{\Delta y^2}{b^2}}$$

where

$$ab = 1, \quad \frac{a}{b} = \frac{\sigma_x}{\sigma_y}$$

We use the Mahalanobis distance in order to prune candidate words and compute the distance between gestures and templates. According to an offline simulation based on the gesture data in Study 1, this modification improved the Top-1 accuracy (predicting the intended word as the most probable) from 81.8% to 84.0% when compared to the Euler distance-based algorithm.

Second, users performed more accurately for the start and end of the gestures than the middle points. This inspired us to assign more weight to the start and end of gestures than to middle points. To determine the optimal weight, we ran a simulation study that sampled weight value from 0 to 1 for the mean distance of middle points. Our results showed that a weight of 0.5 would yield an optimal Top-1 prediction accuracy of 88.8%. This provided the second improvement of the gesture-word recognition algorithm.

$$D_M = 0.25 \times D_{M_1} + 0.25 \times D_{M_N} + 0.5 \times \frac{\sum_{i=2}^{N-1} D_{M_i}}{N-2}$$

where D_{M_i} is the Mahalanobis distance between the ith pair of sampled points of template and input shape.

STUDY 2: THE POTENTIAL OF HEAD GESTURE TYPING

In this experiment, we further investigated *GestureType*. The goal was to explore the expert performance with our modified algorithm.

Participants and Apparatus

We recruited 12 participants (8 males and 4 females; aged from 20 to 24, M = 21.08) in this study. None participated in the first study. Participants rated their familiarities of QWERTY keyboard between 3 and 5 (M = 4.17). Nine participants used gesture typing before but none of them used it as their default keyboard. Eleven participants had previous experience with HMDs.

We used the same apparatus as in Study 1. We employed the modified word-gesture recognition algorithm based on the Mahalanobis distance.

Design

The experiment was designed to have eight sessions. In each session, all participants transcribed the same ten phrases that were randomly sampled from the MacKenzie phrase set [18], which contained a total of 49 words, among which 39 were distinct. On average, a word contained 5.15 (SD = 2.54) letters. In this study, the only independent factor was Session.

Procedure

Before the experiment, we described the goal of the experiment and the interaction method to the participants. Participants then familiarized themselves with the HMD device and interaction for five minutes. We then instructed them to perform "as accurately and quickly as possible". After each session, participants took a 1-minute break. After the experiment, we asked them to fill a questionnaire and interviewed them about the subjective feedback.

Result

Text-Entry Rate

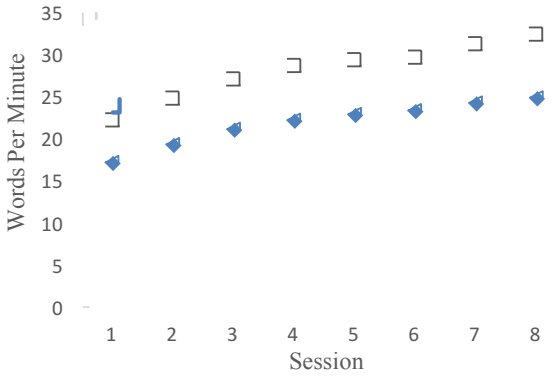

Figure 5: Mean entry rates (WPM) over sessions in Study2. Error bars indicate standard deviation.

As shown in Fig. 5, users typed 17.04 WPM (SD = 6.02) in the first session and improved by 45.13% to achieve 24.73 WPM (SD = 8.48) in the last session. Text entry rate was increased by 5.03 WPM in Session 1-4, and 2.01 WPM in Session 5-8. The best performer typed 25 WPM in the first session and ended up at 39 WPM. The learning curve seemed not to converge at the end of the experiment.

Error Rate

In the eight sessions, the word-level uncorrected error rates were 1.19%-2.56% (M = 1.96%, SD = 0.60%), while the word-level corrected error rates were 1.38%-5.96% (M = 3.86%, SD = 1.35%). There was no significant effect of Session on either error rates.

The learning effect

To gain deeper insight into performance improvement, we broke down the text entry time into three components: gesturing time, selecting time and elapsed time. Gesture time was the time spent on performing gestures. Selecting time was the time span from ending the gesture to clicking on a candidate word. Elapsed time was the time span from inputting the last word to starting the gesture of the next word. According to the data, gesturing time, selecting time and elapsed time accounted for 61.64%, 7.46% and 30.90% of the total text entry time respectively. As shown in Fig. 6, the three component times continued to decrease with sessions, and started to converge in the 5th or 6th session. On the other hand, the Top-1 accuracy improved from 85.0% in the first session to 89.8% in the last session, even though ANOVA showed no significant effect of Session on Top-1 accuracy ($F_{7,77} = 1.69$, $p = .12$). These results showed that participants learned to improve gesturing speed without sacrificing final accuracy.

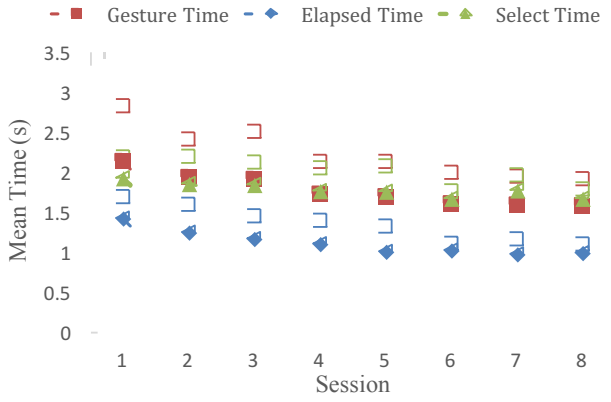

Figure 6: Mean gesturing time, selecting time and elapsed time over sessions in Study 2. Error bars indicate standard deviation.

The post-experiment interview also offered insights into the strategy of users to perform better by adapting their input behavior. As participants reported, they found longer words to be more tolerant to inaccurate gestures. Therefore, they performed gestures faster for longer words. To verify this, we examined how the start distance, the end distance and the middle-point distance changed with word length (the number of contained letters). As shown in Fig. 7, as word length increased, the mean gesture start/end distances were relatively stable while the mean middle-point distance increased significantly ($F_{8,88}=51.9$, $p<.0001$). This indicated that for longer words, participants actually performed more inaccurately for the middle points rather than gesture start and end, in order to save movement time. Moreover, users had learned strategies to input individual words. Take "shopping" for example. Since it was easy to confuse with "shipping", participants reported that they would traverse 'o' carefully to avoid 'i'. Examples also included "breathing" vs. "breaking", "confirm" vs. "conform", etc.

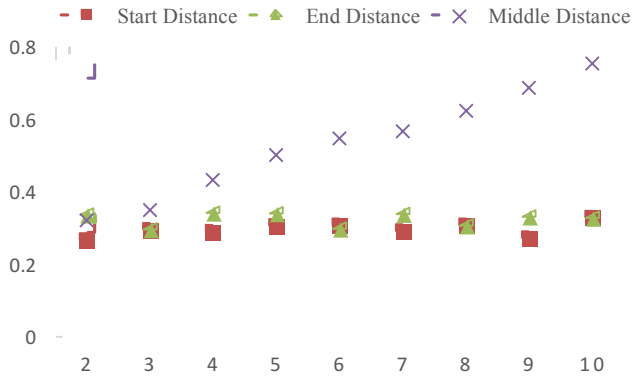

Figure 7: Distance (in key size) of gesture start, gesture end, and middle points vs. word length in Study2.

Improvement of the gesture-word recognition algorithm

We assessed the power of the gesture-word recognition algorithm based on the Top-1 accuracy (predicting users' intended word as the best match). In Study 1 and Study 2, the mean Top-1 accuracies were 81.27% and 84.5% respectively. The improvement was 3.23%, which was not as large as we found in the simulation study (7%). However, we should consider the fact that participants transcribed the same ten phrases in eight times in Study 2, and text entry rate was 29% faster. To enable fair comparison, we ran another simulation by testing the algorithm of Study 1 with gesture data in Study 2. Results of this showed that the Top-1 accuracy was only 74.19%, which was much lower than the 84.5% we found in Study 2. This result showed the improved algorithm indeed had a stronger power to decode gestures into words.

Subjective user feedback

The subjective user feedback was generally the same as that of Study 1 for *GestureType*. In addition, most participants reported they started to feel fatigue at the 6th session, which was after intermittently typing 60 phrases for 40 minutes. This result showed that the fatigue from head-based typing was acceptable when text entry is short on HMDs (e.g. searching a keyword or replying to a message).

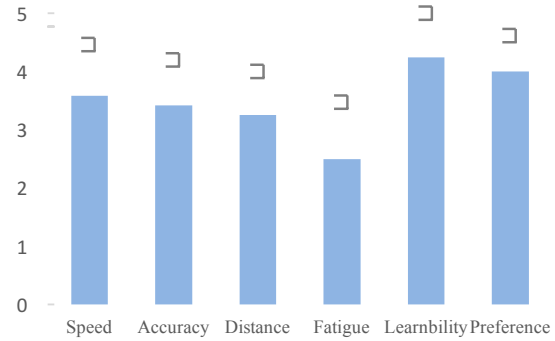

Figure 8: Subjective user feedback of Study 2 with error bars showing standard deviation.

LIMITATION AND FUTURE WORK

The present research has a number of limitations, which point to the direction of future work.

First, in this paper, the keyboard size and the control/ display ratio were determined according to the results of our pilot study. We did not thoroughly research their effects on users' ability of head-based pointing or text entry.

Second, while this research focuses more on *GestureType*, there remains room to research the other two techniques (*DwellType* and *TapType*) in more depth. For example, it would be valuable to research how expert users will adjust the dwell time and how fast they can input with *DwellType*.

Third, *GestureType* only supports words in a predefined dictionary. However, inputting OOV (out of vocabulary) words is also important for practical use. One simple solution is to combine *TapType* with it to support both word-level and character-level input. The mode switch should be smooth if we can distinguish a tap and a gesture accurately. It would then be useful to examine the resulting performance.

Forth, we used simple classical algorithms to parse input in this research. More sophisticated algorithms and models such as LSTM for gesture keyboard decoding [**Error! Reference source not found.**] and HMM decoder [**Error! Reference source not found.**] should further improve the performance. We acknowledge that the obtained performance of this paper does not reflect the ceiling rate, but it is appropriate to explore the feasibility and compare the general performance of three techniques.

Fifth, the current evaluation is a lab study. It is valuable to run a long-term field experiment to test the performance over a longer period of study and whether results would be affected by external environment or everyday composition.

CONCLUSION

Head-mounted displays are emerging interaction platforms, which can accommodate various VR and AR applications. To our knowledge, our research is the first to compare different head-based text entry techniques on HMDs. We tested three representative methods with and without hands. Our results showed that head-based text entry techniques are both feasible and practical solutions on HMDs. Head-based text entry techniques are not as fatiguing as it may seem at first impression. Users can learn to type with their head very quickly, and the text entry speeds are satisfactory.

In particular, we investigated gesture typing on HMDs in depth. Our research was the first to adapt a gesture keyboard to a HMD. We demonstrated that without the need to pause to select keys, a head-based gesture keyboard (*GestureType*) was unexpectedly fast (24.73 WPM), and significantly outperformed head-based tap input (*TapType*) by 59%. In contrast, the difference between the two input methods for finger-based keyboard was not as significant [25]. This is probably because the head is much slower and less flexible than the finger. Therefore, our research contributes a new scenario where gesture keyboards offer an incomparable advantage.

Our research also identified patterns of head movement in HMDs: User performed more accurately on X axis than on Y axis. We utilize this finding to improve the gesture-word recognition algorithm. As a result, both the simulation study and the real-user study demonstrated the validity and strength of the improvement. In this sense, we also contributed a gesture-word recognition algorithm variation, more specifically, by leveraging the Mahalanobis distance. We hope this method can inspire future research about gesture keyboard input where the motor control ability on X-axis and Y-axis are not equal. Moreover, the pattern of head movement we found in HMD may also provide guidance for more HMD research other than text entry.

ACKNOWLEDGEMENT

This work is supported by the National Key Research and Development Plan under Grant No. 2016YFB1001200, the Natural Science Foundation of China under Grant No. 61303076 and No. 61572276, Tsinghua University Research Funding No. 20151080408.

REFERENCES

1. 2011. The Open American National Corpus. http://www.anc.org/
2. Ouais Alsharif, Tom Ouyang, Franc¸oise Beaufays, Shumin Zhai, Thomas Breuel, Johan Schalkwyk. Long short term memory neural network for keyboard gesture decoding. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP). IEEE, 2015. 2076-2080
3. Christoph Amma, Marcus Georgi, and Tanja Schultz. 2012. Hands-Free Mobile Text Input by Spotting and Continuous Recognition of 3d-Space Handwriting with Inertial Sensors. In *2012 16th International Symposium on Wearable Computers* (ISWC'12), 52-59.
4. Xiaojun Bi, Ciprian Chelba, Tom Ouyang, Kurt Partridge, and Shumin Zhai. 2012. Bimanual gesture keyboard. In *Proceedings of the 25th annual ACM symposium on User interface software and technology* (UIST '12). 137-146. http://dx.doi.org/10.1145/2380116.2380136
5. Doug A. Bowman, Vinh Q. Ly, and Joshua M. Campbell. 2001. Pinch keyboard: Natural text input for immersive virtual environments.
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/10.1145/2642918.264735

7. Weiya Chen, Anthony Plancoulaine, Nicolas Férey, Damien Touraine, Julien Nelson, and Patrick Bourdot. 2013. 6DoF navigation in virtual worlds: comparison of joystick-based and head-controlled paradigms. In *Proceedings of the 19th ACM Symposium on Virtual Reality Software and Technology*(VRST '13). 111-114. http://dx.doi.org/10.1145/2503713.2503754
8. Wenxin Feng, Ming Chen, and Margrit Betke. 2014. Target reverse crossing: a selection method for camera-based mouse-replacement systems. In *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments* (PETRA '14). Article 39, 4 pages. http://dx.doi.org/10.1145/2674396.2674443
9. Yulia Gizatdinova, Oleg Špakov, and Veikko Surakka. 2012. Comparison of video-based pointing and selection techniques for hands-free text entry. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (AVI '12), Genny Tortora, Stefano Levialdi, and Maurizio Tucci (Eds.). 132-139. http://dx.doi.org/10.1145/2254556.2254582
10. 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
11. Mitchell Gordon, Tom Ouyang, and Shumin Zhai. 2016. WatchWriter: Tap and Gesture Typing on a Smartwatch Miniature Keyboard with Statistical Decoding. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16). 3817-3821. http://dx.doi.org/10.1145/2858036.2858242
12. Tovi Grossman, Xiang Anthony Chen, and George Fitzmaurice. 2015. Typing on Glasses: Adapting Text Entry to Smart Eyewear. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '15). 144-152. http://dx.doi.org/10.1145/2785830.2785867
13. John Paulin Hansen, Kristian Tørning, Anders Sewerin Johansen, Kenji Itoh, and Hirotaka Aoki. 2004. Gaze typing compared with input by head and hand. In *Proceedings of the 2004 symposium on Eye tracking research & applications* (ETRA '04). 131-138. http://dx.doi.org/10.1145/968363.968389
14. Per Ola Kristensson and Keith Vertanen. 2012. The potential of dwell-free eye-typing for fast assistive gaze communication. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (ETRA '12), Stephen N. Spencer (Ed.). 241-244. http://dx.doi.org/10.1145/2168556.2168605
15. Per Ola Kristensson and Shumin Zhai. 2004. SHARK^2: 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
16. Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto, and Margrit Betke. 2016. EyeSwipe: Dwell-free Text Entry Using Gaze Paths. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16). 1952-1956. http://dx.doi.org/10.1145/2858036.2858335
17. I. Scott MacKenzie and R. William Soukoreff. 2002. A character-level error analysis technique for evaluating text entry methods. In *Proceedings of the second Nordic conference on Human-computer interaction* (NordiCHI '02). 243-246. http://dx.doi.org/10.1145/572020.572056
18. 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
19. Päivi Majaranta and Kari-Jouko Räihä. 2007. Text entry by gaze: Utilizing eye-tracking. *Text entry systems: Mobility, accessibility, universality*: 175-187.
20. Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. 2009. Fast gaze typing with an adjustable dwell time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '09). 357-360. http://dx.doi.org/10.1145/1518701.1518758
21. Anders Markussen, Mikkel R. Jakobsen, and Kasper Hornbæk. 2013. Selection-based mid-air text entry on large displays. In *IFIP Conference on Human-Computer Interaction*. 401-418.
22. 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
23. Roderick McCall, Benoît Martin, Andrei Popleteev, Nicolas Louveton and Thomas Engel. 2015. Text entry on smart glasses. In *2015 8th International Conference on Human System Interaction* (HSI). 195-200.
24. Tao Ni, Doug Bowman, and Chris North. 2011. AirStroke: bringing unistroke text entry to freehand gesture interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11). 2473-2476. http://dx.doi.org/10.1145/1978942.1979303

25. Paul Nation and Robert Waring. 1997. Vocabulary size, text coverage and word lists. *Vocabulary: Description, acquisition and pedagogy Rev* 14: 6-19.
26. Diogo Pedrosa, Maria Da Graça Pimentel, Amy Wright, and Khai N. Truong. 2015. Filteryedping: Design Challenges and User Performance of Dwell-Free Eye Typing. *ACM Trans. Access. Comput.* 6, 1, Article 3 (March 2015), 37 pages. http://dx.doi.org/10.1145/2724728
27. Shyam Reyal, Shumin Zhai, and Per Ola Kristensson. 2015. Performance and User Experience of Touchscreen and Gesture Keyboards in a Lab Setting and in the Wild. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15). 679-688. http://dx.doi.org/10.1145/2702123.2702597
28. Sayan Sarcar, Prateek Panwar, and Tuhin Chakraborty. 2013. EyeK: an efficient dwell-free eye gaze-based text entry system. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction* (APCHI '13). 215-220. http://dx.doi.org/10.1145/2525194.2525288
29. Alexander Schick, Daniel Morlock, Christoph Amma, Tanja Schultz, and Rainer Stiefelhagen. 2012. Vision-based handwriting recognition for unrestricted text input in mid-air. In *Proceedings of the 14th ACM international conference on Multimodal interaction* (ICMI '12). 217-220. http://dx.doi.org/10.1145/2388676.2388719
30. Srinath Sridhar, Anna Maria Feit, Christian Theobalt, and Antti Oulasvirta. 2015. Investigating the Dexterity of Multi-Finger Input for Mid-Air Text Entry. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15). 3643-3652. http://dx.doi.org/10.1145/2702123.2702136
31. Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. VelociTap: Investigating Fast Mobile Text Entry using Sentence-Based Decoding of Touchscreen Keyboard Input. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15). 659-668. http://dx.doi.org/10.1145/2702123.2702135
32. Daniel Vogel and Patrick Baudisch. 2007. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '07). 657-666. http://dx.doi.org/10.1145/1240624.1240727
33. Xin Yi, Chun Yu, Mingrui Zhang, Sida Gao, Ke Sun, and Yuanchun Shi. 2015. ATK: Enabling Ten-Finger Freehand Typing in Air Based on 3D Hand Tracking Data. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (UIST '15). 539-548. http://dx.doi.org/10.1145/2807442.2807504
34. Chun Yu, Ke Sun, Mingyuan Zhong, Xincheng Li, Peijun Zhao, and Yuanchun Shi. 2016. One-Dimensional Handwriting: Inputting Letters and Words on Smart Glasses. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16). 71-82. http://dx.doi.org/10.1145/2858036.2858542
35. Shumin Zhai, Per Ola Kristensson, Pengjun Gong, Michael Greiner, Shilei Allen Peng, Liang Mico Liu, and Anthony Dunnigan. 2009. Shapewriter on the iphone: from the laboratory to the real world. In*CHI '09 Extended Abstracts on Human Factors in Computing Systems* (CHI EA '09). 2667-2670. http://dx.doi.org/10.1145/1520340.1520380
36. Shumin Zhai, Carlos Morimoto, and Steven Ihde. 1999. Manual and gaze input cascaded (MAGIC) pointing. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (CHI '99). 246-253. http://dx.doi.org/10.1145/302979.303053
37. 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
38. Shumin Zhai and Per Ola Kristensson. 2012. The word-gesture keyboard: reimagining keyboard interaction. *Commun. ACM* 55, 9 (September 2012): 91-101. http://dx.doi.org/10.1145/2330667.2330689