iText: Hands-free Text Entry on an Imaginary Keyboard for Augmented Reality Systems

Xueshi Lu
Xi'an Jiaotong-Liverpool University
China
xueshi.lu17@gmail.com

Difeng Yu
The University of Melbourne
Australia
difeng.yu@student.unimelb.edu.au

Hai-Ning Liang*
Xi'an Jiaotong-Liverpool University
China
haining.liang@xjtlu.edu.cn

Jorge Goncalves
The University of Melbourne
Australia
jorge.goncalves@unimelb.edu.au

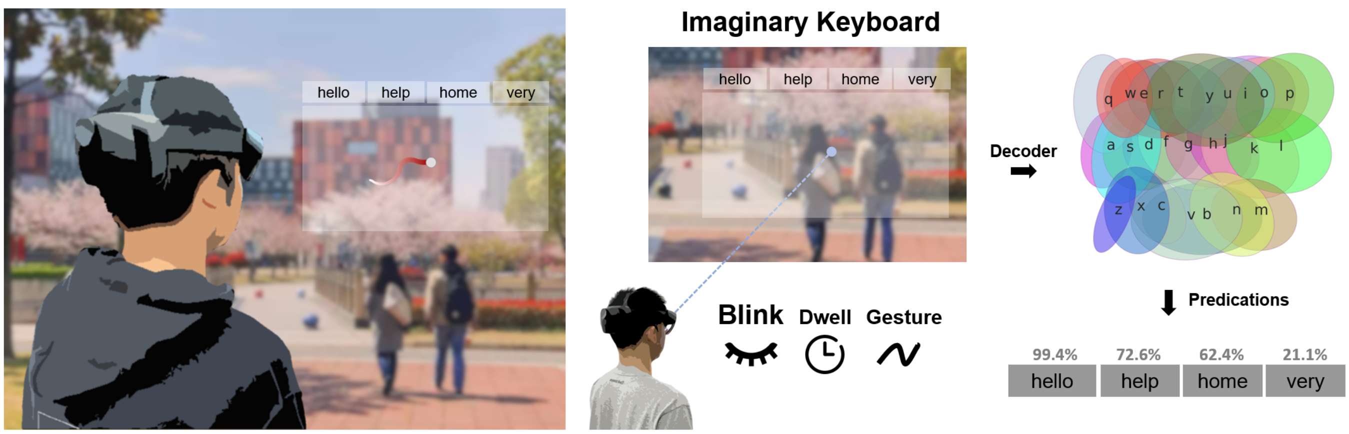

Figure 1: Left: a user is typing on iText in an AR HMD. It uses an imaginary (invisible) keyboard and allows efficient text entry in a hands-free and occlusion-free manner. Right: iText accepts user input through eye blinks, dwell, or head swipe gestures and predicts the intended words from a statistical decoder.

ABSTRACT

Text entry is an important and frequent task in interactive devices including augmented reality head-mounted displays (AR HMDs). In current AR HMDs, there are still two main open challenges to overcome for efficient and usable text entry: arm fatigue due to mid-air input and visual occlusion because of their small see-through displays. To address these challenges, we present iText, a technique for AR HMDs that is hands-free and is based on an imaginary (invisible) keyboard. We first show that it is feasible and practical to use an imaginary keyboard on AR HMDs. Then, we evaluated its performance and usability with three hands-free selection mechanisms: eye blinks (E-Type), dwell (D-Type), and swipe gestures (G-Type). Our results show that users could achieve an average text entry speed of 11.95, 9.03 and 9.84 words per minutes (WPM) with E-Type, D-Type, and G-Type, respectively. Given that iText with E-Type outperformed the other two selection mechanisms in text entry rate and subjective feedback, we ran a third, 5-day study. Our results show that iText with E-Type can achieve an average text entry rate of 13.76 WPM with a mean word error rate of 1.5%. In short, iText can enable efficient eyes-free text entry and can be useful for various application scenarios in AR HMDs.

*The corresponding author.

UIST '21, October 10–14, 2021, Virtual Event, USA

ACM ISBN 978-1-4503-8635-7/21/10…$15.00
https://doi.org/10.1145/3472749.3474788

CCS CONCEPTS

• **Human-centered computing → Interaction techniques**; *Text input*.

KEYWORDS

text entry, typing, hands-free, eye blink, dwell, augmented reality, head-mounted displays

ACM Reference Format:
Xueshi Lu, Difeng Yu, Hai-Ning Liang, and Jorge Goncalves. 2021. iText: Hands-free Text Entry on an Imaginary Keyboard for Augmented Reality

Systems. In *The 34th Annual ACM Symposium on User Interface Software and Technology (UIST '21), October 10–14, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3472749.3474788

1 INTRODUCTION

Augmented reality head-mounted displays (AR HMDs) have become increasingly more practical and appealing due to a reduction in their weight and size, and significant improvements in their display and computing power. They provide virtual overlays that augment the physical world and have the potential to help users complete everyday tasks such as checking emails, browsing information, and online chatting more effectively and conveniently [16]. However, efficient typing (or text entry) remains an important open challenge in these scenarios, one that needs to be solved before this vision can become a reality.

The state-of-the-art AR devices, such as HoloLens 2, require users to perform pinch gestures to select keys on a mid-air keyboard. However, due to the small see-through displays, virtual interfaces (e.g., the virtual keyboard) can occlude users from seeing what objects are behind the interface, either virtual and/or physical [14]. This occlusion and lack of awareness of their surroundings can even cause physical injuries when users are standing and walking [28]. Another challenge with pinch-based approaches is that the hands may be unavailable for typing, because they are occupied with other tasks, such as holding other items. Moreover, mid-air input can easily lead to arm fatigue [10]. Due to these issues, current techniques based on mid-air typing and visible virtual keyboards are difficult to apply to everyday use scenarios for AR devices, which are meant to give users the ability to move around.

The above challenges can be addressed by having a text entry technique that is hands-free and uses an imaginary (invisible) keyboard. Typing on an imaginary keyboard, where the keys are not shown on the display, affords users with almost full visibility of physical environments [38]. Additionally, a technique that is hands-free allows users to type even when their hands are occupied. Therefore, integrating hands-free and an imaginary keyboard is a promising solution for text entry in AR systems.

In this paper, we present iText, a technique that is hands-free and uses an imaginary keyboard for efficient text entry for AR systems. The final implementation of iText is the result of an iterative design process. In a first study, we used a Wizard of Oz approach [3] to explore the feasibility of hands-free text entry on an imaginary keyboard. We explored two types of selection mechanisms: (1) D-Type (short for dwell type) that allows users to select letters discretely by hovering the cursor on the keys, and (2) G-Type (short for gesture type) which allows text entry by making continuous gestures. Our results show that it is feasible to use hands-free input with an imaginary keyboard as users could recall imaginary key positions accurately. Based on the statistical decoders derived from the first study, iText was implemented with three selection mechanisms (D-Type, G-Type, and E-Type), where E-Type (short for eye type) uses eye blinks to trigger selections. E-Type was included since recent research [17] showed that eye blinks are an efficient and usable hands-free technique for virtual reality HMDs. We evaluated their effectiveness and usability through a second user study with 18 participants. Our results show that iText with E-Type, D-Type and G-Type can reach an average text entry speed of 11.95, 9.03 and 9.84 words per minute (WPM). As E-Type showed superior performance and higher user preference, we ran a third 5-day study to understand text entry speed, error rate, learning curve and eye fatigue levels of iText with E-Type based on longer-term use of the technique. Our results show that iText with E-Type achieved an average of 13.76 WPM on the last day of deployment. Furthermore, induced eye fatigue was low, and user performance and usability of iText was barely affected.

The main contributions of the paper include: (1) iText, a novel efficient hands-free text entry technique for AR HMDs that uses an imaginary keyboard; (2) an empirical study to show the feasibility of hands-free text entry on an imaginary keyboard in AR HMDs; (3) an implementation and comparative evaluation of three hands-free selection mechanisms (blinking, dwell, and gesture) on imaginary keyboards; and (4) a long-term evaluation of iText with blinking as input. As a hands-free technique that uses a non-visible keyboard, iText addresses two important issues: occlusion and arm fatigue.

2 RELATED WORK

In this section we present our review of the literature on text entry techniques for AR/VR, invisible/imaginary interfaces, and hands-free approaches for text entry.

2.1 Text Entry Techniques in AR HMDs

Several techniques have been proposed to enable text entry in AR HMDs. For example, VISAR [4] allows users to input text via a visualised input surface with mid-air hand typing. To avoid visual occlusion, VISAR minimizes the visual clutter by only showing the outlines and labels of the keys. In the final experiment, VISAR was reported to achieve an average of 17.75 WPM text entry speed. HIBEY [15] employs a one-line keyboard layout and allows users to browse through or select a key via mid-air gestures, and was proposed as a viable solution for text entry in a constrained AR screen. A long-term user study showed that HIBEY led to a mean typing rate of 9.95 WPM over an eight-day study. Both VISAR and HIBEY were implemented in HoloLens 1, which afforded a limited size of field of view (FOV = 34 degrees [8]). Xu et al. [33] evaluated eight text entry techniques with four input mechanisms (controller, head, hand and hybrid) with two selection methods (tap and swipe) in AR HMDs. The results indicated that controller-based typing techniques were better than other techniques in terms of text entry performance and subjective feedback. However, the above mid-air text entry techniques were reported to have a common issue of hand/arm fatigue during the experiments. To our knowledge, no hands-free text entry technique has been investigated in AR HMDs for scenarios where users' hands are busy with other tasks.

2.2 Hands-free Text Entry Techniques

Speech recognition is a feasible way to achieve hands-free text entry. Ruan et al. [29] used a state-of-art speech recognition system and compared it with a mobile touchscreen typing method. The study revealed that transcribing phrases using speech was nearly three times faster than on a touchscreen keyboard. Although speech allowed speedy text entry in transcribing tasks, Foley et al. [5] stated that speech was no better than typing on a touchscreen for composing tasks, and people were more inclined to choose a touchscreen keyboard according to their subjective feedback. In addition, there are two major disadvantages of speech techniques: (1) they require a relatively quiet environment for speech recognition techniques to work properly; and (2) they can lead to privacy concerns. These issues prevent speech text entry to be used in many public spaces like libraries, restaurants, malls, etc.

Besides speech recognition, a few techniques allow users to enter texts through head/eye movements. Dwell-based input [24, 26] is a prevalent solution for selecting one target key at a time (character-level text entry), in which a user gazes at a target key for a certain time (so called dwell time) to trigger the selection. Eyes-S [27] allows users to draw letters on 9 pre-determined regions based on a 400ms dwell time and enables users to reach a text entry speed of 6.8 WPM on average. Majaranta et al. [20] conducted a long-term user study that allowed the users to personalize the dwell time. Their results showed that the mean typing speed increased to 19.9 WPM after some practice. Mott et al. [22] proposed a dwell-based text entry with a cascading dwell time. This approach led to a mean typing speed of 12.39 WPM. Instead of eye gazing, Yu et al. [37] indicated that it is possible to reach 10.59 WPM with a constant 400ms dwell time for VR HMDs using a head-based typing technique. However, dwell-based approaches have inherent limitations—a long dwell time can decrease user performance and a short dwell time is likely to 'push' users to select keys quickly, which could increase the likelihood of false-positive selections. Dwell-free techniques can overcome the shortcomings of using a dwell time and allow users to have more control of their typing speed. RingText [34], for example, provides users with a circular keyboard to type in a dwell-free fashion in VR HMDs. Experts were able to reach an average typing speed of 13.24 WPM after 90 minutes of training. BlinkType [17] is an alternative dwell-free approach, where users enter text by blinking their eyes. An experiment showed that BlinkType was more efficient (13.47 WPM) and preferred by users compared to a dwell-based approach.

Some dwell-free techniques may also allow word-level text entry (i.e., entering a word each time rather than a character). With EyeSwipe [13], users select the first and last characters of a word by reverse-crossing and glancing through the vicinity of the middle characters. Candidate words were displayed when performing the gesture. EyeSwipe reached a typing rate of 11.7 WPM after 30 minutes of practice. In Filteryedping [25], users looked at characters in the same order of a required word, then the system automatically filtered out unwanted words and ranked candidate words according to their length and frequency. This technique reached a mean text entry speed of 15.95 WPM. In addition to gaze-based typing methods, a head-based technique called GestureType [37] has been shown to achieve a typing speed of 19.04 WPM in VR environments. However, GestureType was not completely hands-free since controller buttons were used to indicate the start and end of drawing.

It should be noted that common eye-tracking systems are highly sensitive to noise, and gaze-based text entry normally suffer from a low typing speed and high error rates. In order to correctly obtain the selections, a gaze-based keyboard usually requires a large key size (e.g., one key occupies about 90px in EyeSwipe [13]). Therefore, controlling the pointer movement through users' head with head-based selection techniques (e.g., head-based dwelling, head-based gestures, head movement + blink selection [17]) can potentially lead to more robust results.

2.3 Imaginary Keyboard

To type on an imaginary keyboard, users have to recall key positions of the QWERTY layout. Zhu et al. [38] investigated an imaginary keyboard on a smartphone, where virtual keys were hidden to save screen space. Users were reported to reach an average text entry speed of 37.9 WPM after 3 days of practice. Their results indicated that users could recall key positions correctly, and the typing paradigm based on imaginary keyboards was easy to learn. Imaginary keyboards have also been explored in eyes-free conditions, where users enter text without looking at the keyboard while receiving text feedback from a distant display. BlindType [18] leverages the thumb's muscle memory to type on a touchpad and can lead to a typing rate of 17-23 WPM. Their work also reported that a classical decoding algorithm was capable of hands-free text entry. i'sFree [39] is a word-level eyes-free text entry technique. It decodes gestures drawn on a remote control device and was reported to reach a text entry speed of 23 WPM. Xu et al. [35, 36] presented text entry on a fingertip keyboard using thumb-tip gestures. With a bimanual fingertip keyboard, this technique approached an average typing speed of 23.4 WPM. From our literature review, we argue that the imaginary keyboard can be successfully adapted from smartphones to AR HMDs without losing any of its remaining advantages.

2.4 Summary

Based on our review, we highlight that dwell-based and dwell-free gesture-based selection mechanisms are viable candidates for hands-free text entry on imaginary keyboards based on a QWERTY layout. Therefore, dwell (D-Type) and gesture (G-Type) were the two selection mechanisms chosen for the feasibility study. We then introduced eye blinking (E-Type) as an alternative input mechanism for discrete, character-level text entry (as an alternative to the dwell-based approach) because of the high usability and efficiency of using eye blinks for character selection and the relatively cheap cost for eye blink detection (rather than gaze tracing) [17].

3 DESIGN OF ITEXT

3.1 Keyboard Layout

The keyboard was placed one meter away at the center of the user's view. The size of the keyboard was 40*20cm and one key-width was set as 3.4cm. The keys were hidden and the panel was 30% transparent, which could be adjusted based on user preference. The last row of the keyboard is slightly more visible for indicating (from left to right) a space key, delete key and function key for moving to the next phrase. The cursor is controlled by users' head movements. The most likely suggested letter predicted via our statistical decoder appears close to the pointer.

3.2 The Three Selection Mechanisms: D-Type, G-Type, and E-Type

These three selection mechanisms were inspired from our literature review. D-Type allows users to enter text by hovering the cursor over a key for a predefined dwell time. We conducted a pilot test

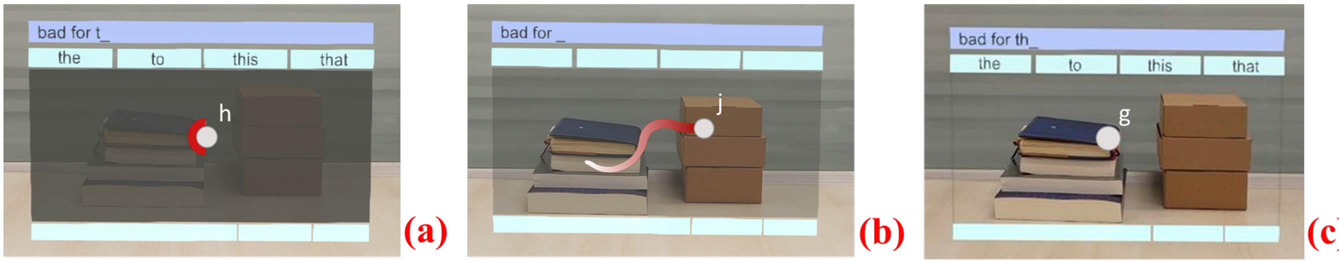

Figure 2: Three illustrations (from left to right) that represent D-Type, G-Type and E-Type, the three techniques that enable hands-free text entry on an imaginary keyboard in AR environments. (a) The user dwells the pointer over a location for a certain time to trigger a selection. (b) The user draws the letters in sequence to enter a word. (c) The user blinks their eyes to trigger a selection. Note that the transparency level of the imaginary keyboard area does not affect the functionality of iText: (a) and (b) show two levels of transparency while in (c) the keyboard area is fully transparent.

to determine a suitable range of dwell time. At the end, we set the dwell time to be 600ms which represented an appropriate trade-off between speed and unintentional selections. A red circular progress bar (Figure 2.a) is used to indicate the dwell time, and a light beep sound to help users know whether the character has been selected. One issue for dwell-based text entry is that users may dwell on the same key after selecting it as they search for the next key. Because of this, dwell time is set to be 800ms after a key is activated. The dwell time refreshes if the cursor moves more than 1.7cm, which is suitable according to our pilot studies.

G-Type lets users enter text by drawing a path through letters of a word in order. G-Type is semi dwell-based, which means the gesture starts or ends from a cursor hovering on a certain place for a dwell time. To be consistent with D-Type, a dwell time of 600ms indicates the start and 800ms indicates the end of the gesture. In addition to a red circle that represents dwell time and a beep sound to indicate a selection, a short red tail is used to help users visualize the cursor movement (Figure 2.b). The dwell time refreshes once the cursor moves more than 1.7cm. Furthermore, users can select a single letter or a suggested word by keeping the pointer on a target place (within 1.7cm) during the period of drawing a gesture.

E-Type allows users to enter text using eye blinks. A recent paper [17] showed that blinking with both eyes resulted in much higher accuracy (~100%) compared to left eye blinking (79.5%) and right eye blinking (69.4%). Therefore, E-Type also uses the blinking of both eyes to trigger selections. We set 400ms as the minimum gap time between double-blinking entries. To avoid false-positive selections caused by unintended eye blinks, users are allowed to look at any space outside the keyboard to reset their eyes temporarily (Figure 2.c). Eye trackers produced by different vendors provide distinct mechanisms to detect eye blinks. In this research, we use a HoloLens 2 which supports an API to identify if the headset is receiving the eye tracking data successfully. We enforced a rule that the system recognizes an eye blink once the headset loses eye tracking data (which happens when eyes are closed) for more than 150ms.

4 USER STUDY 1

This experiment aimed to investigate typing behaviors of D-Type and G-Type on an imaginary keyboard. It was carried out with three goals: 1) To understand whether users can transfer their memory of the location of the keys to AR environments and type with an imaginary keyboard in AR HMDs; 2) To understand whether users can perform hands-free text entry on an imaginary keyboard; and 3) To capture and compare endpoint distributions of D-Type and G-Type.

4.1 Participants and Apparatus

We recruited 16 participants aged from 19 to 23. The average score of the familiarity with the QWERTY keyboard layout (1: novice; 5: expert) was 4, with a minimum of 3. The average score of confidence of typing without looking at the keyboard (1: not confident; 5: confident) was 3, only one participant gave a rating of 1. Nine participants also reported having limited exposure to AR HMDs. The experiment was conducted with a HoloLens 2, which afforded a diagonal field of view (FOV) of 52 degrees. We developed the experimental environment with Unity3D.

4.2 Experiment Design and Procedure

Since users perform different typing patterns under gesture-like (G-Type) and tap-like (D-Type and E-Type) methods according to prior studies [18, 39], we only captured typing behaviors from two input modalities (D-Type and G-Type) in this study. That is, we did not include E-Type because D-Type and E-Type were likely to have similar endpoint distributions, given that they are both characterized by head pointing and character-level entry (i.e., text entry is done character by character). We designed a Wizard of Oz experiment to collect users' unbiased typing behaviors, and no decoding algorithms were employed. Wizard of Oz is an experimental approach in which participants interact with a computer system (in our case, via a keyboard interface) that they believe to be autonomous and functional but it is actually pre-defined or operated by an unseen human [3]. Participants were required to transcribe phrases using D-Type or G-Type. The system always displayed the correct text no matter where the pointer hit.

The experiment followed a within-subjects design. The independent variable was Selection Mechanism (D-Type and G-Type). Their order was counterbalanced to avoid carryover effects. At the beginning of the experiment, each participant was briefed with the details of the selection mechanism and the AR HMD they would use. Participants were instructed to complete two tasks, one for each

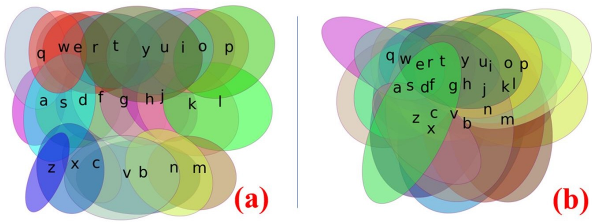

Figure 3: Imaginary key distributions on D-Type (a) and G-Type (b) covered by 95% confidence ellipses.

mechanism. Participants were given 5 phrases to get them familiar with each of the two mechanisms. We instructed participants to complete the warm-up session or until they got comfortable using the keyboard, which happened at a different pace for different participants. Each task consisted of 4 blocks, and each block had 6 phrases. The phrases were randomly generated from the MacKenzie phrase set [19]. Participants could take a short break between blocks and review the real QWERTY keyboard if they wanted to. Participants were instructed to provide the endpoints as accurately and naturally as possible. The experiment took around 50 minutes for each participant.

4.3 Results

4.3.1 Data Pre-processing. It is not straightforward to obtain imaginary key positions directly from a gesture which is made up of continuous points. We employed the method used in i'sFree [39] to infer users' imaginary key positions. Their method was based on a dynamic time warping (DTW) algorithm [1]. The key idea is to compare the input gesture with the templates from the corpus (both sampled into 100 equidistant points), and then find the optimal match. The imaginary key has the shortest DTW distance to the corresponding key position in the template.

We used endpoint distribution to illustrate the results, which is formed by multiple intersection points of the selection ray (as controlled by the head) and the virtual keyboard when a key selection happens. The endpoints made by D-Type and G-Type can be regarded as "Touch Down/Up" and "Swipe" events on a touchscreen. At the end, we collected a total of 20370 number of endpoints from the experiment, 9397 with D-Type and 10973 with G-Type. In order to reduce the effect of outliers, we dropped the points that are far away (more than three standard deviations) from the key centroids. This process led to the removal of 1354 outliers (6.65%).

4.3.2 Variances. Figure 3 shows that the mean standard deviations across imaginary keys, which are 3.94cm and 2.44cm in the x-axis, and 2.90cm and 1.46cm in the y-axis for G-Type and D-Type, respectively. One-way ANOVA tests showed that Selection Mechanism had significant effects on mean standard deviations ($F_{(1,50)} = 36.056, p < 0.001$ for the x-axis; $F_{(1,50)} = 69.092, p < 0.001$ for the y-axis). A repeated-measures ANOVA with Greenhouse-Geisser correction showed that the mean standard deviations significantly differed between the axes ($F_{(1,50)} = 115.176, p < 0.001$). The mean standard deviation in the x-axis was 0.99cm greater than the y-axis.

4.3.3 Offsets. The horizontal (x-axis) and vertical (y-axis) offsets were calculated by $(x_t, y_t) - (x_v, y_v)$, where (x_t, y_t) represents the coordinate of the imaginary key position, and (x_v, y_v) is the coordinate of the target key. The positive and negative value of the offset indicate that the imaginary key is located above or below the key center. The mean horizontal offset across keys is -1.65cm (SD = 1.30) for D-Type and -3.31cm (SD = 2.20) for G-Type. D-Type and G-Type yielded 0.16cm (SD = 0.46) and 2.71cm (SD = 1.45) vertical offsets, respectively. One-Way ANOVA tests revealed that selection methods had a significant effect on offsets ($F_{(1,50)} = 10.921, p < 0.002$ for the x-axis; $F_{(1,50)} = 72.974, p < 0.001$ for the y-axis).

4.3.4 Discussion. During the experiment, we found that most of the participants did not complete the practice sessions and went directly to the formal task; in particular, they practiced fewer phrases in D-Type compared to G-Type. Moreover, most participants did not need to review the layout of the keys on a real keyboard after the second break. The confidence ellipses revealed that the centers of the imaginary keys almost match those of the keys in a standard QWERTY keyboard layout. This shows that participants could recall key positions on the imaginary keyboard in AR environments (*Goal 1*). Similarly, the results show that they can perform hands-free text entry on the imaginary keyboard based on the QWERTY layout (*Goal 2*). We also observed that G-Type resulted in larger mean standard deviations across keys, which was about 1.5cm greater than D-Type in both axes. These results indicate that D-Type is more stable with smaller standard deviation of key positions (*Goal 3*). Despite this, the mean standard deviation of G-Type is about one key-width, which is still acceptable. The offsets showed that imaginary keys with the two typing mechanisms were likely to be located on the upper left of the key centers (*Goal 3*). Our results also show that G-Type shifted nearly one-key width to the left and D-Type barely shifted in the y-axis.

4.4 Deriving Typing Decoders

We used the data to implement a statistical decoder with an adapted spatial model to suggest words to the users. Normally, the statistical decoder [6] consists of a spatial model which gives the probability distributions with regard to key locations on the keyboard, and a language model that decides the prior probabilities of words based on the corpus. We adapted a spatial model to fit the imaginary keyboard [38]. At the end, the decoder could infer the most likely letters according to the spatial model once the user triggers a selection according to the following equation:

$$C^* = \arg\max_{C \in A} P(x, y|C) \tag{1}$$

where A is comprised of 26 letters of the English alphabet on the keyboard, (x, y) represents the endpoint coordinate the user hits on the keyboard, C is one character belonging to the alphabet, and C^* is the predicted character. Equation 1 was used to produce the most likely character under the current selection endpoint (as shown in Figure 2). Then, we assume that each selection is independent and $P(x, y|C)$ follows a bivariate Gaussian distribution. Based on this, the decoder suggests the words with high probabilities calculated with the following equation:

$$W^* = \arg\max_{W \in L} P(S|W)P(W) \tag{2}$$

where L is the language corpus which has a lexicon of 10k words from the American National Corpus [11], W represents a particular word in the corpus, and S is the input. In this case, $P(S|W)$ is calculated from the adapted spatial model with parameters collected from Study 1 and $P(W)$ is from a language model.

For G-Type, since a dwell-based like mechanism is used to indicate the start/end of a gesture, we employed Equation 1 to recognize the corresponding letters. We also applied the gesture recognition algorithm reported in previous research [12, 21, 37]. The algorithm infers the most probable word by measuring the similarities of the shapes. The words in the corpus are transformed into templates in the form of lines that connect the center of each letter of a word in order. Then, the input gesture and templates are sampled into 100 equidistant points. The algorithm prunes the corpus by removing the templates whose start/end positions are more than 1-key width away from the input's start/end positions. The similarity is calculated by the sum of the Euler distance between points on the gesture and the template. The word that leads to the minimum Euler distance is the best candidate word based on the gesture.

5 USER STUDY 2

We conducted a second study to evaluate the effectiveness of the three hands-free selection mechanisms on an imaginary keyboard. Our aim was to explore typing speed, error rate, learning rate, and subjective feedback and usability issues.

5.1 Participants and Apparatus

We recruited another 18 participants from a university campus, 6 for each technique. The participants were aged between 19 to 23 years (mean=21). No participant was involved in the previous study. The average score of their familiarity with QWERTY keyboards (1: naive; 5: expert) was 4. The average score of confidence of typing without looking at the keyboard (1: not confident; 5: confident) was 3. The experiment was conducted with the same device as Study 1.

5.2 Experiment Design and Procedure

The experiment followed a between-subjects design to avoid cross-learning effects. Participants were asked to provide their demographic information before the experiment. Then, they were briefed with the details of the hands-free text entry mechanisms and imaginary keyboard interface. At the beginning of the task, participants were given 5 phrases to get them familiar with the system. Then, a total of 40 phrases were randomly generated for each participant from the MacKenzie phrase set [19]. The phrases were divided into 5 blocks evenly. Participants were allowed to have a short break between the blocks. After they completed the task, participants were asked to fill out a NASA-TLX [9] and a short version of User Experience Questionnaire (UEQ-S) [31] to provide their subjective feedback. The whole experiment took about 50 minutes per participant.

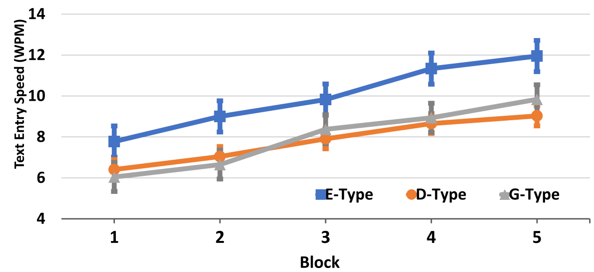

Figure 4: Mean text entry speeds of E-Type, D-Type and G-Type across 5 blocks. Error bars indicate the standard error.

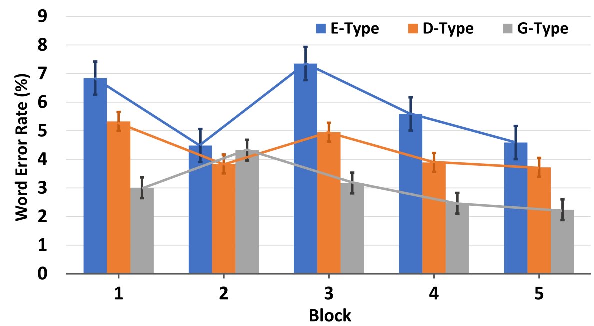

Figure 5: Word error rates of E-Type, D-Type and G-Type across 5 blocks. Error bars indicate the standard error.

5.3 Results

To analyze the data collected in this study, we employed repeated-measures ANOVAs (RM-ANOVAs) and Bonferroni-adjusted pairwise comparisons with Block (blocks 1-6) as the within-subjects factor and Selection Mechanism (D-Type, G-Type and E-Type) as the between-subjects factor.

Text entry speed was measured in Words Per Minutes (WPM):

$$WPM = \frac{|S-1|}{T} \times \frac{1}{5} \tag{3}$$

where S is the length of the transcribed string in characters and T is the elapsed time in minutes timed from the first keystroke to the last keystroke for one phrase.

The word error rate [2, 32, 38] was measured by Minimum Word Distance (MWD) because the keyboard employed word-level corrections. The MWD was the smallest number of word deletions, insertions, or replacements needed to transform the transcribed string into the desired string. The word error rate is defined as:

$$r = \frac{MWD(S,P)}{|P|} \times 100\% \tag{4}$$

where $MWD(S,P)$ represents the MWD between the transcribed phrase S and the target phrase P, and $|P|$ denotes the number of words in P.

5.3.1 Text Entry Speed. RM-ANOVA tests yielded a significant effect of Block ($F_{(4,60)} = 88.385, p < 0.001$) and Selection Mechanism ($F_{(2,15)} = 6.829, p < 0.008$) on text entry speed. E-Type was the fastest mechanism as pairwise comparisons showed that

Table 1: UEQ scales in terms of pragmatic, hedonic and overall quality.

Technique	Pragmatic	Hedonic	Overall
E-Type	1.67 (Good)	1.29 (Above Average)	1.48 (Good)
D-Type	1.21 (Above Average)	1.04 (Below Average)	1.13 (Above Average)
G-Type	0.38 (Bad)	0.88 (Below Average)	0.63 (Below Average)

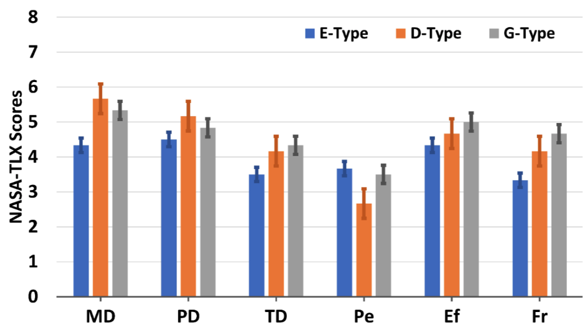

Figure 6: NASA-TLX ratings of E-Type, D-Type and G-Type in terms of Mental Demand (MD), Physical Demand (PD), Temporal Demand (TD), Performance (Pe), Effort (Ef) and Frustration (Fr). Error bars indicate the standard error.

E-Type was significantly faster than D-Type ($p = 0.014$) and G-Type ($p = 0.023$). Participants achieved an average of 7.77 WPM ($SD = 1.45$) in the first block, and the typing speed increased to 11.95 WPM ($SD = 1.30$) in the final block. D-Type was faster than G-Type in the first two blocks. However, G-Type achieved an average of 9.84 WPM ($SD = 1.37$) in the final block, which was faster than D-Type ($M = 9.03$ WPM, $SD = 1.15$).

5.3.2 Error Rate. The word error rate was low in the final block among the three mechanisms, with E-Type, D-Type and G-Type having 4.6%, 3.7% and 2.3% respectively. G-Type yielded a relatively low error rate through the five blocks; it dropped below 3.0% in the last two blocks. With E-Type, the error rate dropped from the first block of 6.9% to the fifth block of 4.6%. The mean error rate of D-Type in the first block was 5.3%, and decreased to 3.7% in the last block. RM-ANOVA tests did not reveal a significant effect on word error rate ($F_{(2,15)} = 2.215, p = 0.144$).

5.3.3 Subjective Feedback. One-way ANOVAs showed that Selection Mechanism had a significant main effect on the "Complicated-Easy" subscale of UEQ ($F_{(2,17)} = 13.280, p < 0.001$). Post-hoc tests showed that the G-Type was significantly more complicated than E-Type ($p < 0.001$). No other significant effects were found. Table 1 shows the summary of UEQ-S ratings. E-Type was considered as the most pragmatic and hedonic mechanism compared to the other two. G-Type was rated lower in the pragmatic value than E-Type.

Figure 6 shows the NASA-TLX subscale ratings for Mental Demand (MD), Physical Demand (PD), Temporal Demand (TD), Performance (Pe), Effort (Ef) and Frustration (Fr). One-way ANOVAs did not show any significant effect of Selection Mechanism on MD ($p = 0.267$), PD ($p = 0.632$), TD ($p = 0.494$), Pe ($p = 0.368$), Ef ($p = 0.753$), or Fr ($p = 0.234$).

5.4 Discussion

Results from this study confirm the feasibility of the three hands-free text entry approaches (that is, dwell-based, dwell-free and gesture-based) with an imaginary keyboard.

E-Type is the most promising solution in terms of text entry speed and user experience feedback. Compared to G-Type and D-Type, E-Type is around 2-3 WPM faster. G-Type and D-Type have similar text entry speeds. E-Type is faster probably because it does not require any dwell time before a selection. While there was no significant difference of Selection Mechanism on word error rate, G-Type showed a slight advantage in error rate while E-Type led to a slightly higher error rate. This is because G-Type requires users to type by selecting suggested words other than character-by-character which offers an alternative way to enter text in characters. This mechanism forces users to be conscious of selecting correct words. Unintended eye blinks are probably the main reason that made E-Type have a slightly higher error rate.

The subjective feedback shows that E-Type had a higher preference score, based on both pragmatic and hedonic measures. G-Type was rated low in pragmatic because of its complexity. Though participants successfully recalled key positions of the QWERTY keyboard and results from prior research [39] have shown that a G-Type selection on an imaginary keyboard using hands was faster than on a visible keyboard, our results show that drawing a word trace based on head motions is not easy for users as the head is not a common input method and is not as natural and conventional as using hands.

The results also show that the typing speed of G-Type was lower compared to previous studies (e.g., 15.58 WPM in [37]). This could be due to the following reasons: (1) The HoloLens 2 used in this experiment affords only half the FOV of a Samsung Gear VR (52 vs 96 degrees), which resulted in a much smaller keyboard size and constrained the speed of drawing traces. (2) Participants could have faced additional challenges when they had to continuously recall key positions in real-time when performing head-based gestures. (3) Our implementation of G-Type is fully hands-free where the start/end of a gesture is decided by a dwell time. On the other hand, the head-based gesture typing in the previous study [37] is not entirely hands-free given that the start/end of the head gesture is triggered by pressing a button using hands.

Because E-Type showed an obvious advantage in text entry speed and user preference, we wanted to explore further its performance over a longer use period and see how long users would need to reach peak entry rates and whether there could be eye fatigue and its effect on performance and usability.

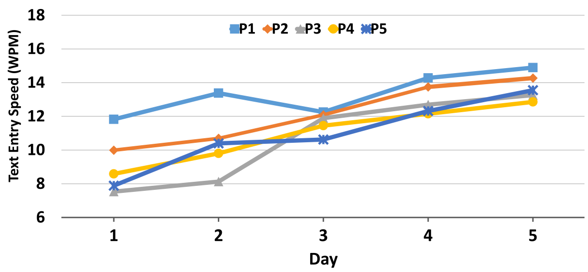

Figure 7: Means of text entry speed of 5 participants across 5 days with E-Type.

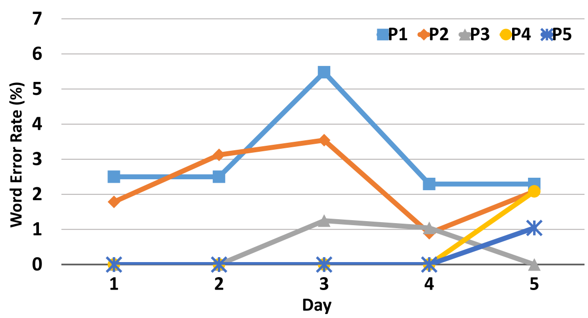

Figure 8: Word error rates of 5 participants across 5 days with E-Type.

6 USER STUDY 3

We conducted a third 5-day user study to examine the long-term usage of iText with E-Type as its only selection mechanism. Since Study 2 revealed that E-Type was the most efficient and preferred by participants, this last study aimed to explore the potential performance (in terms of typing speed and word error rate), learning rate and eye fatigue of iText with E-Type. Five participants were recruited from the same campus, aged between 20 to 22 years (mean=21). These participants were not involved in the previous two studies. The average score of their familiarity with QWERTY keyboards (1: naive; 5: expert) was 4. The average score of confidence of typing without looking at the keyboard (1: not confident; 5: confident) was 3. The apparatus used was the same as in the two previous studies.

6.1 Experiment Design and Procedure

The study consisted of 5 continuous days. Participants were instructed to complete two blocks, each block consisted of 8 phrases randomly generated from the MacKenzie phrase set [19]. Participants were able to have a rest between blocks. Over 5 days, participants were given two phrases to warm up before the actual task and were asked to rate their eye fatigue (1: very low; 7: very high) after the task.

6.2 Results

6.2.1 Text Entry Speed. A one-way ANOVA revealed a significant main effect of Day on text entry speed ($F_{(4,49)} = 17.018, p < 0.001$). Post-hoc comparisons showed significant differences between Day 1 *vs.* Day 3 ($p = 0.003$), Day 1 *vs.* Day 4 ($p < 0.001$), Day 1 *vs.* Day 5 ($p < 0.001$), Day 2 *vs.* Day 4 ($p = 0.002$), Day 2 *vs.* Day 5 ($p < 0.001$), Day 3 *vs.* Day 5 ($p = 0.020$). Participants achieved average text entry rates of 13.04 and 13.77 WPM on Day 4 and Day 5, respectively.

6.2.2 Word Error Rate. There was no significant effect of Day on word error rate ($F_{(4,49)} = 0.677, p = 0.611$). Overall, the word error rate was low for most of the study. All participants had an error rate under 3.0% in the last two days ($M = 0.9\%$ in Day 4, $M = 1.5\%$ in Day 5). While P1 ($M = 3.0\%, SD = 1.2\%$) and P2 ($M = 2.3\%, SD = 0.9\%$) had a relatively higher error rate compared to the other participants over 5 days, they performed text entry with high speed ($M = 13.33$ and $M = 12.16$, respectively).

6.2.3 Subjective Ratings. One-way ANOVA tests revealed a main effect of Day on eye fatigue ($F_{(4,24)} = 4.875, p = 0.007$). Post-hoc comparisons showed that there is a significant difference between Day 1 and Day 5 ($p = 0.005$). The mean rates dropped 60% from Day 1 ($M = 3.00, SD = 1.0$) to Day 5 ($M = 1.20, SD = 0.45$). 4 participants rated 1 in eye fatigue in the last day. The results indicated that eye fatigue decreased along with users getting familiar with the blinking mechanism.

7 DISCUSSION AND POTENTIAL APPLICATION SCENARIOS

The results of the three studies lead to three main findings. First, the distributions (Figure 3) of imaginary key positions indicate that users can transfer their memory of the QWERTY keyboard to an imaginary keyboard in AR. They can recall the positions of keys correctly with both discrete selection (D-Type) and continuous selection (G-Type) mechanisms. While dwell-based selection shows a closer match of key locations with small standard deviations, gesture-based selection is still viable on an imaginary keyboard. This result paves the path for exploring further imaginary keyboards in AR HMDs (and potentially for VR HMDs as well). Second, E-Type outperforms D-Type and G-Type in terms of text entry speed and subjective feedback. While our results show that E-Type is practical and easy to learn and use, D-Type and G-Type are both practical and feasible mechanisms for text selection on imaginary keyboards. Third, a 5-day user study shows that E-Type can achieve a text entry rate of 13.77 WPM with a word error rate below 3%. Moreover, self-reported ratings indicate that entering text via eye blinks only causes slight eye fatigue (4 of 5 participants rated eye fatigue as very low in the last day). While E-Type is promising and works very well with iText, it requires an eye tracker which may not be available for all AR HMDs. However, eye tracking technology is becoming cheaper [7, 30] and is integrated into more AR HMDs including HoloLens 2 and Magic Leap, and also VR HMDs such as HTC VIVE Pro Eye, HP Reverb G2 Omnicept Edition, and Pico Neo 2 Eye. As such, we believe that eye tracking will likely be a standard feature of these devices. Overall, we recommend that E-Type gets priority for iText type of text entry techniques when eye tracking is available; otherwise, D-Type and G-Type are feasible alternatives when eye trackers are not available.

We envision that iText could be applied to various scenarios. Below, we propose three potential application scenarios.

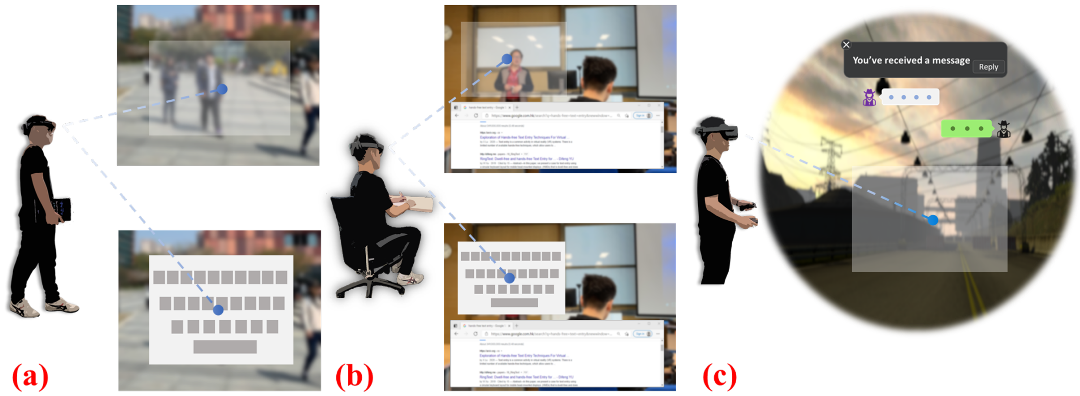

Figure 9: Three potential application scenarios for iText. (a) a user is walking with his hands occupied. With iText he could enter text while walking and still seeing the surrounding objects clearly. (b) a user is sitting and making a concept map while listening to a lecture. He could use iText to search for a key concept on a search engine without losing sight of the speaker. (c) a user is playing a game in a VR HMD and a message comes in. He could reply quickly using iText without stopping the game.

- Mobile augmented reality is intended to be used for interacting with mid-air interfaces on the go. Some existing research has explored the use of AR HMDs while users are walking [14, 23]. Both of these studies have indicated issues of users lacking awareness of their physical surroundings. By leveraging the advantages of hands-free and an imaginary keyboard interface, iText could be an effective solution for text entry while users are walking while at the same time being able to see clearly nearby surrounding objects, which should lead to a safer and more comfortable experience (see Figure 9.a).
- iText makes it possible to achieve text entry on an overlapping interface and as such it is suitable for switching among interfaces in a more seamless way, while still having a clear view of the environment. For example, let's imagine that a user is attending a lecture talk. With their hands occupied making a concept map of the content of the talk, they want to search for a definition of a concept that the lecturer is talking about. With a standard keyboard, it will be likely that the keyboard will cause occlusion. On the other hand, with iText, the occlusion will be reduced and the user could type in the search keyword and still be able to have a good view of the lecturer and the hall (see Figure 9.b).
- iText could also be implemented in VR HMDs. As using eye blinks as a hands-free selection mechanism has been shown to work well in VR HMDs already [17] and these devices can capture head motions, iText in VR can provide a similar level of performance and user experience. iText in VR can open the door for some interesting uses and possibilities not possible with a visible keyboard, for example allowing users to play games and respond to text messages without significantly interrupting the game (see Figure 9.c) .

8 LIMITATIONS AND FUTURE WORK

Our research has some limitations which can represent directions for future work. First, we did not explore how keyboard size affects the text entry performance and usability. Smaller key sizes could speed up the movement of the pointer across keys. However, because the experiments were already lengthy (~ 50 minutes), adding more independent variables would introduce fatigue issues. Therefore, our work focused on comparing the performance of the three hands-free selection mechanisms with a single keyboard size. Future work can explore different keyboard sizes which may affect user performance. Moreover, even with the state-of-art AR HMDs, their FOV is limited. As such, it is difficult to have large or small keyboards in these devices. When newer HMDs come with expanded FOV, issues related to key sizes and gaps between keys could be explored further for non-visible keyboards. Second, although the keyboard area in the experiments is shown as transparent, it was only tested with a clean background inside a closed door lab environment (see Figure 2). It will be useful to explore the performance of iText in other environmental backgrounds and with users performing other tasks (like walking or running).

9 CONCLUSION

In this paper, we have presented iText, a novel hands-free text entry technique on an imaginary keyboard for augmented reality head-mounted displays (AR HMDs). Being hands-free and using an imaginary keyboard overcomes usability issues related to arm fatigue due to mid-air input and occlusion of objects (both virtual and physical) because of their small see-through display. To develop iText, we first showed via a user study the feasibility of using hands-free for text entry on an imaginary keyboard, as users were able to recall key positions with relatively high accuracy. From the data collected, we were able to then derive adapted statistical decoders suggesting words in iText. In our second study, we evaluated the performance of iText with three hands-free selection mechanisms: E-Type that uses eye blinks, D-Type that is based on dwell, and G-Type that relies on swipe gestures. Our results show that users could achieve average text entry rates of 11.95, 9.03 and 9.84 WPM, respectively. A third, five-day study with iText plus E-Type showed that users can achieve an average typing speed of 13.76 WPM with a low word error rate and negligible eye fatigue. Overall, iText is a novel technique that allows for efficient hands-free text entry via an imaginary keyboard and has great potential to be applied in mobile AR HMDs and be integrated with other applications.

ACKNOWLEDGMENTS

The authors want to thank the participants for their time and the reviewers for their helpful feedback. The work was supported in part by Xi'an Jiaotong-Liverpool University (XJTLU) Key Special Fund (#KSF-A-03).

REFERENCES

[1] Donald J. Berndt and James Clifford. 1994. Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 359–370.

[2] Xiaojun Bi, Shiri Azenkot, Kurt Partridge, and Shumin Zhai. 2013. Octopus: evaluating touchscreen keyboard correction and recognition algorithms via. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 543–552. https://doi.org/10.1145/2470654.2470732

[3] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. 1993. Wizard of Oz studies—why and how. *Knowledge-based systems* 6, 4 (1993), 258–266. https://doi.org/10.1016/0950-7051(93)90017-N

[4] John J. Dudley, Keith Vertanen, and Per Ola Kristensson. 2018. Fast and Precise Touch-Based Text Entry for Head-Mounted Augmented Reality with Variable Occlusion. 25, 6 Interact. (2018), Article 30. https://doi.org/10.1145/3232163

[5] Margaret Foley, Géry Casiez, and Daniel Vogel. 2020. Comparing Smartphone Speech Recognition and Touchscreen Typing for Composition and Transcription. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 1–11. https://doi.org/10.1145/3313831.3376861

[6] 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*. Association for Computing Machinery, 194–195. https://doi.org/10.1145/502716.502753

[7] Simone Guasconi, Marco Porta, Cristiano Resta, and Carlo Rottenbacher. 2017. A low-cost implementation of an eye tracking system for driver's gaze analysis. In *2017 10th International Conference on Human System Interactions (HSI)*. IEEE, 264–269. https://doi.org/10.1109/HSI.2017.8005043

[8] Ramy Hammady and Minhua Ma. 2019. *Designing spatial ui as a solution of the narrow fov of microsoft hololens: Prototype of virtual museum guide*. Springer, 217–231. https://doi.org/10.1007/978-3-030-06246-0_16

[9] Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 50. Sage publications Sage CA: Los Angeles, CA, 904–908. https://doi.org/10.1177/154193120605000909

[10] Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, and Pourang Irani. 2014. Consumed endurance: a metric to quantify arm fatigue of mid-air interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 1063–1072. https://doi.org/10.1145/2556288.2557130

[11] Nancy Ide and Catherine Macleod. 2001. The american national corpus: A standardized resource of american english. In *Proceedings of corpus linguistics*, Vol. 3. Citeseer, 1–7.

[12] Per-Ola Kristensson and Shumin Zhai. 2004. SHARK: a large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*. Association for Computing Machinery, 43–52. https://doi.org/10.1145/1029632.1029640

[13] 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*. Association for Computing Machinery, 1952–1956. https://doi.org/10.1145/2858036.2858335

[14] Wallace S. Lages and Doug A. Bowman. 2019. Walking with adaptive augmented reality workspaces: design and usage patterns. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. Association for Computing Machinery, 356–366. https://doi.org/10.1145/3301275.3302278

[15] Lik Hang Lee, Kit Yung Lam, Yui Pan Yau, Tristan Braud, and Pan Hui. 2019. Hibey: Hide the keyboard in augmented reality. In *2019 IEEE International Conference on Pervasive Computing and Communications (PerCom*. IEEE, 1–10. https://doi.org/10.1109/PERCOM.2019.8767420

[16] Feiyu Lu, Shakiba Davari, Lee Lisle, Yuan Li, and Doug A Bowman. 2020. Glanceable AR: Evaluating Information Access Methods for Head-Worn Augmented Reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 930–939. https://doi.org/10.1109/VR46266.2020.00113

[17] Xueshi Lu, Difeng Yu, Hai-Ning Liang, Wenge Xu, Yuzheng Chen, Xiang Li, and Hasan Khalad. 2020. Exploration of Hands-free Text Entry Techniques for Virtual Reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 344–349. https://doi.org/10.1109/ISMAR50242.2020.00061

[18] Yiqin Lu, Chun Yu, Xin Yi, Yuanchun Shi, and Shengdong Zhao. 2017. BlindType: Eyes-Free Text Entry on Handheld Touchpad by Leveraging Thumb's Muscle Memory. 1, 2 Wearable Ubiquitous Technol. (2017), Article 18. https://doi.org/10.1145/3090083

[19] 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*. Association for Computing Machinery, 754–755. https://doi.org/10.1145/765891.765971

[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*. Association for Computing Machinery, 357–360. https://doi.org/10.1145/1518701.1518758

[21] 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*. Association for Computing Machinery, 1073–1082. https://doi.org/10.1145/2556288.2556964

[22] Martez E. Mott, Shane Williams, Jacob O. Wobbrock, and Meredith Ringel Morris. 2017. Improving Dwell-Based Gaze Typing with Dynamic, Cascading Dwell Times. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2558–2570. https://doi.org/10.1145/3025453.3025517

[23] Florian Müller, Martin Schmitz, Daniel Schmitt, Sebastian Günther, Markus Funk, and Max Mühlhäuser. 2020. Walk The Line: Leveraging Lateral Shifts of the Walking Path as an Input Modality for Head-Mounted Displays. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 1–15. https://doi.org/10.1145/3313831.3376852

[24] Christian Müller-Tomfelde. 2007. Dwell-based pointing in applications of human computer interaction. In *IFIP Conference on Human-Computer Interaction*. Springer, 560–573. https://doi.org/10.1007/978-3-540-74796-3_56

[25] Diogo Pedrosa, Maria Da Graça Pimentel, Amy Wright, and Khai N. Truong. 2015. Filteryedping: Design Challenges and User Performance of Dwell-Free Eye Typing. 6, 1 Comput. (2015), Article 3. https://doi.org/10.1145/2724728

[26] Abdul Moiz Penkar, Christof Lutteroth, and Gerald Weber. 2012. Designing for the eye: design parameters for dwell in gaze interaction. In *Proceedings of the 24th Australian Computer-Human Interaction Conference*. Association for Computing Machinery, 479–488. https://doi.org/10.1145/2414536.2414609

[27] Marco Porta and Matteo Turina. 2008. Eye-S: a full-screen input modality for pure eye-based communication. In *Proceedings of the 2008 symposium on Eye tracking research & applications*. Association for Computing Machinery, 27–34. https://doi.org/10.1145/1344471.1344477

[28] Romain Pourchon, Pierre-Majorique Léger, Élise Labonté-LeMoyne, Sylvain Sénécal, François Bellavance, Marc Fredette, and François Courtemanche. 2017. Is augmented reality leading to more risky behaviors? An experiment with Pokemon Go. In *International Conference on HCI in Business, Government, and Organizations*. Springer, 354–361. https://doi.org/10.1007/978-3-319-58481-2_27

[29] Sherry Ruan, Jacob O. Wobbrock, Kenny Liou, Andrew Ng, and James A. Landay. 2018. Comparing Speech and Keyboard Text Entry for Short Messages in Two Languages on Touchscreen Phones. 1, 4 Wearable Ubiquitous Technol. (2018), Article 159. https://doi.org/10.1145/3161187

[30] Thiago Santini, Wolfgang Fuhl, David Geisler, and Enkelejda Kasneci. 2017. EyeRecToo: Open-source Software for Real-time Pervasive Head-mounted Eye Tracking. In *12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*. 96–101. https://doi.org/10.5220/0006224700960101

[31] Martin Schrepp, Andreas Hinderks, and Jörg Thomaschewski. 2017. Design and Evaluation of a Short Version of the User Experience Questionnaire (UEQ-S). *Ijimai* 4, 6 (2017), 103–108. https://doi.org/10.9781/ijimai.2017.09.001

[32] R. William Soukoreff and I. Scott MacKenzie. 2001. Measuring errors in text entry tasks: an application of the Levenshtein string distance statistic. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems*. Association for Computing Machinery, 319–320. https://doi.org/10.1145/634067.634256

[33] Wenge Xu, Hai-Ning Liang, Anqi He, and Zifan Wang. 2019. Pointing and selection methods for text entry in augmented reality head mounted displays. In *2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 279–288. https://doi.org/10.1109/ISMAR.2019.00026

[34] Wenge Xu, Hai-Ning Liang, Yuxuan Zhao, Tianyu Zhang, Difeng Yu, and Diego Monteiro. 2019. Ringtext: Dwell-free and hands-free text entry for mobile head-mounted displays using head motions. *IEEE transactions on visualization and computer graphics* 25, 7, 2513–2513. https://doi.org/10.1109/TVCG.2019.2913518

[35] Zheer Xu, Weihao Chen, Dongyang Zhao, Jiehui Luo, Te-Yen Wu, Jun Gong, Sicheng Yin, Jialun Zhai, and Xing-Dong Yang. 2020. BiTipText: Bimanual Eyes-Free Text Entry on a Fingertip Keyboard. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 1–13. https://doi.org/10.1145/3313831.3376306

[36] Zheer Xu, Pui Chung Wong, Jun Gong, Te-Yen Wu, Aditya Shekhar Nittala, Xiaojun Bi, Jürgen Steimle, Hongbo Fu, Kening Zhu, and Xing-Dong Yang. 2019. TipText: Eyes-Free Text Entry on a Fingertip Keyboard. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*. Association for Computing Machinery, 883–899. https://doi.org/10.1145/3332165.3347865

[37] Chun Yu, Yizheng Gu, Zhican Yang, Xin Yi, Hengliang Luo, and Yuanchun Shi. 2017. Tap, Dwell or Gesture? Exploring Head-Based Text Entry Techniques for HMDs. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 4479–4488. https://doi.org/10.1145/3025453.3025964

[38] Suwen Zhu, Tianyao Luo, Xiaojun Bi, and Shumin Zhai. 2018. Typing on an Invisible Keyboard. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, Paper 439. https://doi.org/10.1145/3173574.3174013

[39] Suwen Zhu, Jingjie Zheng, Shumin Zhai, and Xiaojun Bi. 2019. i'sFree: Eyes-Free Gesture Typing via a Touch-Enabled Remote Control. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, Paper 448. https://doi.org/10.1145/3290605.3300678