2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)

Pointing and Selection Methods for Text Entry in Augmented Reality Head-Mounted Displays

Wenge Xu*
Xi'an Jiaotong-Liverpool University

Hai-Ning Liang †‡
Xi'an Jiaotong-Liverpool University

Anqi He§
The University of Adelaide

Zifan Wang¶
Xi'an Jiaotong-Liverpool University

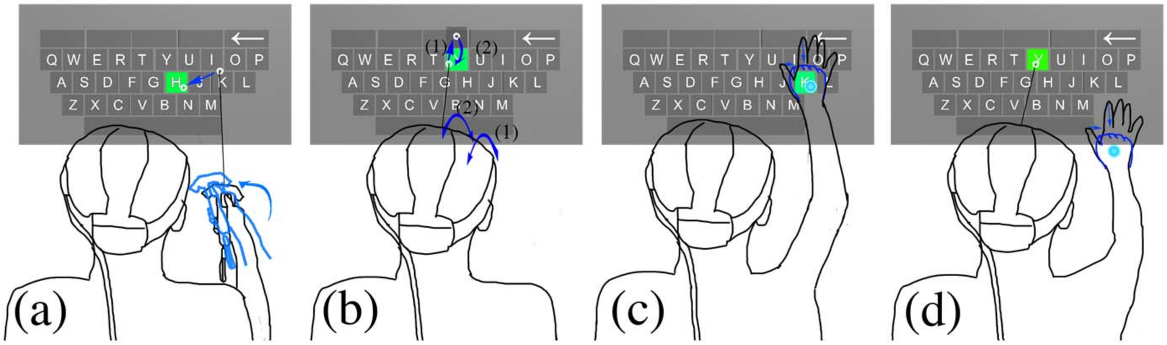

Figure 1: This figure shows how to Tap a letter by using the four pointing methods for AR HMDs. (a) *Controller*—the user uses a controller to move the cursor on the letter 'H', and then presses the trigger button to confirm the selection; (b) *Head*—the user positions the cursor on the letter 'Y', waits for 500ms for the popup button to appear, then (1) moves the cursor to the popup button, and (2) returns to the letter 'Y' to select it; (c) *Hand*—the user moves the hand to the letter 'K' and makes a close palm gesture to selects it; (d) *Hybrid (Head+Hand)*—the user uses the head to move the cursor to the letter 'Y' and makes a close palm gesture to make the selection.

ABSTRACT

Augmented reality (AR) is on the rise with consumer-level head-mounted displays (HMDs) becoming available in recent years. Text entry is an essential activity for AR systems, but it is still relatively underexplored. Although it is possible to use a physical keyboard to enter text in AR systems, it is not the most optimal and ideal way because it confines the uses to a stationary position and within indoor environments. Instead, a virtual keyboard seems more suitable. Text entry via virtual keyboards requires a pointing method and a selection mechanism. Although there exist various combinations of pointing+selection mechanisms, it is not well understood how well suited each combination is to support fast text entry speed with low error rates and positive usability (regarding workload, user experience, motion sickness, and immersion). In this research, we perform an empirical study to investigate user preference and text entry performance of four pointing methods (*Controller, Head, Hand, and Hybrid*) in combination with two input mechanisms (*Swype and Tap*). Our research represents a first systematic investigation of these eight possible combinations. Our results show that Controller outperforms all the other device-free methods in both text entry performance and user experience. However, device-free pointing methods can be usable depending on task requirements and users' preferences and physical condition.

Keywords: Augmented Reality, Text Entry, User Performance, User Preference, Pointing Methods, Selection Mechanisms

Index Terms: Human-centered computing—Mixed/augmented reality; Human-centered computing—Text input; Human-centered computing—Pointing; Human-centered computing—Gestural input

1 INTRODUCTION

Text entry is an essential activity in all interactive systems, including virtual reality (VR) and augmented reality (AR). There have been some advances in this area for VR [18, 20, 42, 43], but it is still quite underexplored for AR. Unlike VR, AR users can see through the transparent head-mounted display (HMD) and it is possible to access a physical keyboard. For example, the HoloLens can connect to a wireless physical keyboard. However, traditional input devices such as mice and keyboards are not suitable for outdoor environments, as they require a type of flat surface to operate on [37]. Moreover, AR HMDs are meant to be mobile devices that enable users to move within both indoor and outdoor environments [8, 25]. Therefore, using a physical keyboard can be useful for text entry in VR settings [18] as the VR HMDs are commonly used in indoor scenarios, but it is unlikely the most suitable way for AR HMDs.

Text entry in AR differs from VR in many aspects. The hand representation can be hidden or virtually presented in VR [17] but not for AR HMDs. There are some known issues that only exist in AR, including layer interference, color blending problem, and layout foreground-background. These issues affect the text readability, visibility, depth ordering, object segmentation, and scene distortion [21] and make it difficult for users to acclimate to the content viewed through see-through displays [31]. Since the text and the virtual keyboard are typically viewed in a fixed location within an HMD screen, other people and objects in the background can become noise and hinder accomplishing various tasks, including entering text.

Early work has investigated using a glove for AR HMDs to interact with the system to support direct manipulation of virtual objects, interaction with symbolic data (e.g., text entry), and doing military

---

*e-mail: wenge.xu@xjtlu.edu.cn
†e-mail:haining.liang@xjtlu.edu.cn
‡Corresponding Author: Hai-Ning Liang
§e-mail:anqi.he@student.adelaide.edu.au
¶e-mail:zifan.wang16@student.xjtlu.edu.cn

978-1-7281-4765-9/19/$31.00 ©2019 IEEE
DOI 10.1109/ISMAR.2019.00026

Table 1: Overview of text entry methods that have already been evaluated in VR that can potentially be used in AR (adapted from [35]): (1) *hands-only*, (2) *head-only*, (3) *hybrid*, (4) *controller*.

Pointing Method	Input Method	Qwerty	Eyes-free	Hands	Haptic Feedback	Potential Device-free for Current AR HMDs	WPM in VR	WPM other
(1)	Soft button selection	✓	✗	1-2	✗	✗	4-7 [16]	33-36 [4]
(2)/(3)/(4)	Mid-air pointing	✓	✗	1-2	✗	(✓)	15.4 [35]	13-19 [29,33]
(2)/(3)	Head pointing	(✓)	✗	0-1	✗	✓	10-15 [41,42]	4.5 [12]
(1)	Gamepad	✗	(✓)	2	(✓)	✗	8-15 [43]	6-7 [39]
(1)/(3)	Physical keyboard	✓	(✓)	1-2	✓	✗	24-67 [20,24]	45-67 [20]
(1)	Finger gestures	✗	✓	1-2	✗	(✓)	6 [16]	22-29 [36]
(1)	Chording	✗	✓	1	✓	✗	3 [16]	47 [26]
(1)	Multi-tap	✗	✓	1	✓	✗	12 [16]	20 [26]

logistics tasks in both indoor and outdoor settings [37]. However, current AR HMDs do not come with an expensive glove specially designed to support such interactions. On the other hand, pointing methods are not only low-cost but can also be used in both indoor and outdoor scenarios. In addition to head-based pointing, other methods rely on the user's hand or involve a handheld device for cursor positioning. Pointing methods are widely used in both VR and AR HMDs and as such it is worth exploring their suitability and relative performance with virtual keyboards. In this research, our primary goal is to explore pointing methods in AR that can work with a virtual keyboard and does not rely on specialized peripheral devices (i.e., Chord [26]) that typically do not come with the AR HMDs.

Our exploration considers three user case scenarios.

(1) When users have access to a ray-casting handheld device which are inexpensive. The assumption is that the users have access to a controller that can interact with an AR environment using ray-casting, a technique commonly used in VR and is also available in AR [3]. For example, the Magic Leap 1 provides a handheld controller that uses this technique.

(2) Hand-based but device-free. There are two scenarios in this condition. (a) Hybrid interaction, which relies on the use of the head to position the cursor pointer on the letters of the keyboard and the hand to trigger their selection. This approach has been used partially in some AR HMDs like HoloLens. (b) Hand-based interaction, which only relies on mid-air hand motions to move a pointer over the letters and a hand gesture to indicate their selection. This approach has been used partially in the Meta 2 and it is thought to be one of the most natural selection methods used to interact with an AR environment [28].

(3) Both device-free and hands-free. This represents the cases where no device is available, and it is based on head motions only for positioning the cursor and making letter selections. It is suitable for cases where users cannot use their hand or lift it comfortably (e.g., a user using AR HMD seating on a chair inside a bus that has limited space or with their hands encumbered because they are holding other objects). This is suitable also for environments that are too noisy for hand tracking (e.g., a user using the AR HMD while walking within a shopping mall because there are likely other moving objects in the background).

In short, we are comparing four standard, common AR pointing methods: *Head, Hand, Hybrid* (i.e., Head plus Hand like what HoloLens uses), and *Controller*. We also want to test two of the most common input mechanisms for making selections, *Tap* and *Swype* (more details to be provided later). Both pointing methods or input mechanisms have been partially studied for VR HMDs (e.g., [22,35,42]) but, to our knowledge, not for AR HMDs. Therefore, we want to compare 8 text entry combinations of pointing methods and input/selection mechanisms for text entry with respect to their performance, error rates, and user preferences. The results of our experiment with 24 participants (12 using Swype and 12 Tap) show that text entry performance of the Controller is comparable to other works in VR [35,42] and non-VR [12,29]. When compared with all the three device-free pointing techniques, the Controller approach outperforms them in text entry performance and leads to better overall user experience. Our results also show that Swype is as fast as Tap but causes lower uncorrected errors even for users who are new to Swype. On the other hand, these two input mechanisms do not show any significant difference in terms of a user's text entry experience, feeling of immersion, motion sickness, and most NASA TLX workload subscales. Finally, Swype is found to cause a heavier temporal workload and frustration than Tap.

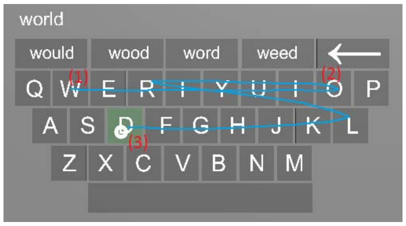

Figure 2: For (Controller/Hand/Head)+Swype, to type the word 'world' a user needs to follow these three steps: (1) Moving the cursor to the first letter 'W' and performing a selection action to indicate the start of the Swype process; (2) then Swyping the letters one by one; and (3) Performing another selection action on the last letter (in this case 'D') to indicate the end.

Table 1 reviews examples of text entry techniques from other domains and devices that could be tailored for AR HMDs. To our knowledge, there has been no study that has explored text entry performance and user experience for AR HMDs. Our study represents the first systematic study of the 8 possible combinations of pointing methods and selection mechanisms. As such, the main contributions of this work include: (1) a first evaluation of four pointing methods × two selection mechanisms (that is, 8 possible combinations) for text input in AR HMDs regarding performance and user preference; and (2) a set of design recommendations that are derived from our experimental results and observations during the experiment.

2 EVALUATED TEXT ENTRY TECHNIQUES

In this section, we describe how each combination of four pointing methods (Controller, Head, Hand, and Hybrid) and two input mechanisms (Tap and Swype) are operationalized in our experiment.

2.1 Controller

One of the most common ways of interacting with virtual environments and their objects is via a handheld controller [38]. The device uses a ray cast from it to the virtual environment to serve as a pointing mechanism. The end of the ray is akin to a cursor. To implement it, we have adapted the HTC VIVE controller (ray-casting enabled

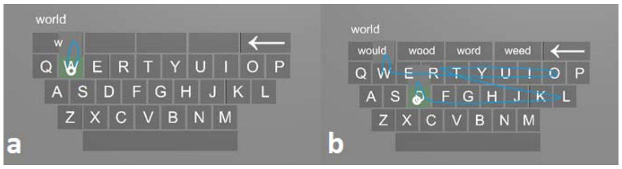

Figure 3: An example of typing the letter ’w’ (a) and the word ’world’ (b) in the Head approach.

with at least one active button) and used the SteamVR Unity plugin to enable it to work with an AR HMD. The users would type on a virtual keyboard by merely moving the controller to point to the desired letters (see Fig. 1a). Selection is done by either Tap or Swype.

Controller+Tap. To select a letter, the user needs to move the cursor to the letter on the virtual keyboard and press the trigger button for selection (see Fig.1a). A Tap action is also required to select a recommended word and special characters (e.g., space/backspace).

Controller+Swype. To type a word, the user needs to move the cursor to the first letter of the intended word and then click the trigger button on the controller to start the Swype action. When the user finishes Swyping, clicking the trigger button again ends the typing process. For special characters, the user needs to move the cursor to the corresponding block and then clicks the trigger button for selection. Fig. 2 shows an example of a Swype action.

2.2 Head

Head-based pointing (or simply Head) is analogous to the Controller, but instead of a handheld device, only the HMD is used. A ray is extended from the HMD position towards the viewing direction into the virtual environment. The ray intersects the keyboard at a point and a blue cursor is given as a prompt (see Fig. 1b).

Head+Tap. Fig. 1b shows an example of how a user completes a Head+Tap action. To enter a word, the user needs to move the cursor using their head to the corresponding letter. A letter selection is made via an outside-inside fashion [22] like a nod action. To begin the process, the user moves the cursor to the target letter; then a button representing an action appears above the letter after a wait time of 500 ms. The user now needs to move the cursor to the button and after moves it back to the target to perform the selection (see Fig. 3a). The user needs to do the action for selecting each letter, suggested word, space, and backspace.

Head+Swype. Selection is like Head+Tap. To type a word, the user needs to perform the selection action on the first letter, then moves the cursor over the component letters, and finally finishes typing by doing the second selection action on the last letter (see Fig. 3b).

2.3 Hybrid

Head-based Pointing + Hand gesture (or simply Hybrid) is a HoloLens-like text input approach. Both implementations of Hybrid+Tap and Hybrid+Swype are analogous to the Head+Tap and Head+Swype, respectively. The only difference is that Hybrid uses a hand gesture (like a palm closing) to indicate a selection.

2.4 Hand

This approach enables users to interact with the virtual keyboard with their hands only. The positions of the palm and hand gestures (i.e., grabbing) are captured via the front camera of the HMD. That is, we use the palm mid-air position to indicate the cursor’s position that acts as the hand-based ’pointing’ (or simply Hand). Users move the cursor according to their hands around the virtual keyboard (see Fig. 1c).

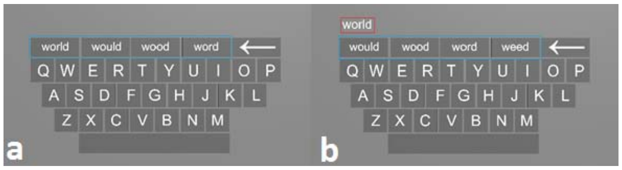

Figure 4: The blue areas show the word suggestions for Tap (a) and Swype (b). In addition, for Swype, the best matched word is automatically added into the input field (the red area).

Hand+Tap. Fig. 1c shows how a user completes a Hand+Tap. Selection is indicated by a palm closing gesture. The user selects a letter by moving the cursor using their hand to the corresponding letter and then selects it by doing a palm closing action. The user should do this to select either a letter, suggested word, or space/backspace. Either left or right hand can be used in this method.

Hand+Swype. Selection is analogous to Hand+Tap. To Swype a word, the user needs to do a first selection gesture on the initial letter of the word to indicate the start, then moves the cursor over the other letters, and finally needs to do the second selection gesture on the last letter to indicate the end of the Swype process. To select a suggested word, delete a letter or add a space, the user needs to move the cursor to the corresponding area, and then do the selection gesture.

2.5 Commonalities and Differences between Swype and Tap

When entering text, it is common for the system to suggest some recommended words based on the typed letters. We have also included the use of these suggested words. Both Swype [14] and Tap (using Symspell [10]) used Damerau-Levenshtein distance algorithm and the same library [1]; as such, the word suggestion performance should not affect the text entry performance.

For Tap, because we do not know whether the user has finished entering the word, we cannot automatically add the best suggestion word into the sentence. All word suggestions appear in the selection blocks (see Fig. 4a, on top of the keys). They are updated every time the user makes a change (i.e., adding or deleting a letter). To select a suggested word, the user needs to choose it from the corresponding selection block. Hitting the space key will append a space after the input. Backspace deletes the last input, which can be a complete word or a single letter.

For Swype, since there is a second selection action to indicate the end of entering a word, the system automatically adds the best word suggestion into the text field with four other possible words in the selection blocks (see Fig. 4b). If the best word suggestion is the intended word, the user can confirm it by Swyping on the next word. If the best suggestion is not the intended word, the user selects the desired word from the selection blocks. The system also automatically appends a space after a word has been input. A delete action deletes the whole word that is last entered.

3 EMPIRICAL STUDY

We conducted an experiment at a university lab with the four pointing methods (Head, Hand, Hybrid, and Controller) and two input mechanisms (Swype and Tap) to assess their relative performance (speed and error rates) and user preference (workload, motion sickness, user experience, and immersion level).

3.1 Participants and Apparatus

24 unpaid participants (8 males and 4 females in each of the two groups) between the ages of 18-28 (M=21) were recruited randomly from the local university campus through a database of participants. All participants were familiar with the English alphabet because the

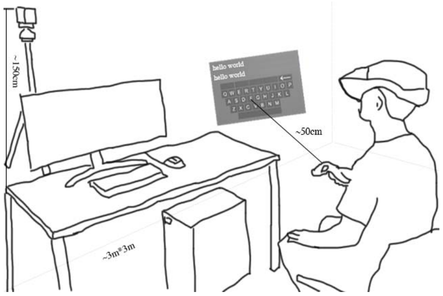

Figure 5: This figure shows the experimental setup. The HTC Vive optical trackers were placed at 1.5m high and had a tracking space with 3 × 3m^2. The keyboard is roughly 0.5m away from the participant which is recommended by the developers of the Meta 2.

language of instruction at the university in English but there were not native alphabet users—English was not their first language. 19 participants had some limited experience with AR HMDs—they had either seen and/or interacted with them. They all had normal or corrected-to-normal vision and did not have any difficulties moving their arms and heads. The experiment was conducted using a Meta 2 AR HMD connected to a Windows 10 machine running Unity3D. A standard desktop computer was used; it had an i7 CPU, 16 GB RAM and a Nvidia GeForce GTX 1080Ti GPU. Figure 5 shows the experimental setup.

3.2 Design

The experiment followed a mix design approach, with one between-subjects independent variable, Input Mechanisms (Swype and Tap), and one within-subjects independent variable, Pointing Methods (Head, Hand, Hybrid, and Controller). The dependent variables were performance (speed and accuracy) and users' subjective feedback (workload, motion sickness, user experience, immersion). Each Input Mechanism was tested on 12 participants (that is, 12 for Swype and 12 for Tap). For each Pointing Method, participants needed to complete 8 phrases which were randomly sampled from the MacKenzie phrase set [27]. To avoid learning effects, we counterbalanced the Pointing Methods. Aside from training phrases, we collected 768 trials (24 participants × 4 Pointing Methods × 8 phrases).

3.3 Procedure

To ensure that both groups have equal text entry ability in the actual experiment stage, participants were separated into two groups (Swype and Tap) based on their performance on a standard desktop PC from a pre-test. Before the experiment, participants were told the goal of the investigation and the conditions that were to be tested. The order of the conditions was balanced across participants. In all conditions, participants were instructed to enter the text phrases as quickly and as accurately as possible. Error correction was allowed by using the backspace key. Before each condition, the Pointing Method was explained to the participants and they practiced two warm-up phrases. After the warm-up phrases, participants needed to complete eight phrases for each condition. The conditions were separated by a 5-minute break during which participants filled out the NASA TLX questionnaire [19], Motion Sickness Assessment Questionnaire (MSAQ) [11], Slater-Usoh-Steed Questionnaire (SUS), and User Experience Questionnaire (UEQ) [23]. After the experiment, we interviewed participants and asked them to comment on the techniques. The whole experiment lasted approximately one hour for each participant.

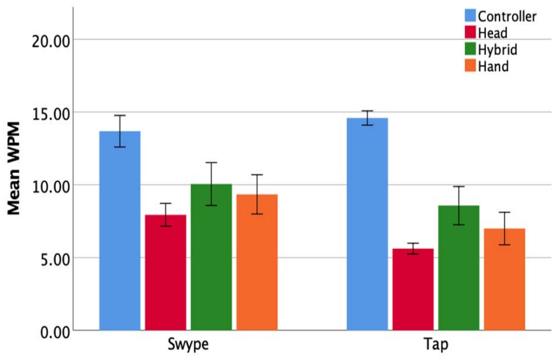

Figure 6: Mean WPM for each Pointing Method grouped by Swype and Tap. Error bars indicate ± 2 standard errors.

3.4 Results

We analyzed the data using a two-way mixed ANOVA with Pointing Methods (Controller, Head, Hand, and Hybrid) as the within-subjects variable and Input Mechanisms (Swype and Tap) as the between-subjects variable. Bonferroni correction was used for pairwise comparisons and Greenhouse-Geisser adjustment was used for degrees of freedom for violations of sphericity. Because of our sample size, the significance threshold was set at $p < .01$ in our analyses.

Text entry rate was measured in Words Per Minute (WPM), with a word defined as five consecutive letters, including the space character. For Swype, we use the following formula

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

Where S was the time (in seconds) from the time when the user triggered the first start action to the last action. $|T|$ was the number of characters in the transcribed text.

For Tap, we use the following formula

$$WPM = \frac{|T|-1}{S} \times 60 \times \frac{1}{5} \tag{2}$$

Where S was the time (in seconds) from the time of the first to the last key entered, and $|T|$ was the number of characters in the transcribed text.

The error rate was calculated based on the standard typing metrics [34], where the total error rate (TER) = not corrected error rate (NCER) + corrected error rate (CER).

3.4.1 Text Entry Performance

Table 2 shows the results from the 2-way mixed ANOVA. Figure 6 shows the mean text entry speed among the 8 techniques. In general, for Pointing Method, Controller achieved the best results for both Tap (M = 14.6, SD = 0.85) and Swype (M = 13.68, SD = 1.88) and Head had the worst performance in both Tap (M = 5.62, SD = 0.64) and Swype (M = 7.94, SD = 1.36). Figure 7 shows the details of the TER and NCER for all methods. Hand caused the highest error rates in TER for both Tap (M=6.48%, SD = 1.80%) and Swype (M = 5.01%, SD = 4.70%) as well as NCER again for both Tap (M = 3.82%, SD = 2.04%) and Swype (M = 0.75%, SD = 0.92%). Head+Tap achieved the lowest TER (M = 1.06%, SD = 1.23%) and NCER (M = 0.48%, SD = 0.82%) while Controller+Swype achieved the lowest TER (M = 1.24%, SD = 1.44%) and NCER (M = 0.00%, SD = 0.00%).

To see if there was significant effect of Pointing Methods for either Tap or Swype, we employed a one-way repeated ANOVA.

Table 2: Two-way mixed ANOVA test results for text entry performance. Significant results where $p < .01$ are shown in green and $p < .001$ in dark green.

	WPM	TER	NCER
Pointing Methods	$F_{2.247,49.428} = 125.890, p < .001$	$F_{3,66} = 15.798, p < .001$	$F_{3,66} = 11.760, p < .001$
Pointing Methods × Input Mechanisms	$F_{2.247,49.428} = 7.225, p < .01$	$F_{3,66} = 2.468, p = .083$	$F_{3,66} = 9.174, p < .001$
Input Mechanisms	$F_{1,22} = 5.227, p = .032$	$F_{1,22} = .055, p = .817$	$F_{1,22} = 18.623, p < .001$
Post-hoc Pointing Methods	Controller - Head ($p < .001$), Controller - Hybrid ($p < .001$), Controller - Hand ($p < .001$), Head - Hybrid ($p < .001$)	Controller - Head ($p < .01$), Head - Hybrid ($p < .01$), Head - Hand ($p < .01$)	N/A

Table 3: ANOVA test results for UEQ subscales. Significant results where $p < .01$ are shown in green and $p < .001$ in dark green. Novelty, Stimulation, Input Mechanisms, Pointing Methods × Input Mechanisms have no significant result and therefore not shown for better clarity.

	Efficiency	Perspicuity	Dependability	Attractiveness
Pointing Methods	$F_{2.244,49.357} = 10.141, p < .001$	$F_{3,66} = 16.170, p < .001$	$F_{3,66} = 5.054, p < .01$	$F_{3,66} = 10.701, p < .001$
Post-hoc Pointing Methods	Head - Controller ($p < .001$), Hand - Controller ($p < .01$), Hybrid - Controller ($p < .01$)	Head - Controller ($p < .001$), Hand - Controller ($p < .001$), Hybrid - Controller ($p < .001$)	Hybrid - Controller ($p < .01$)	Hand - Controller ($p < .001$), Hybrid - Controller ($p < .01$)

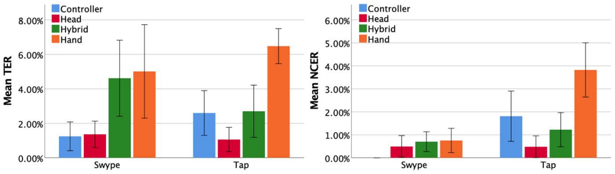

Figure 7: Mean TER (a; left) and NCER (b; right) in % for all methods. Error bars indicate ± 2 standard errors.

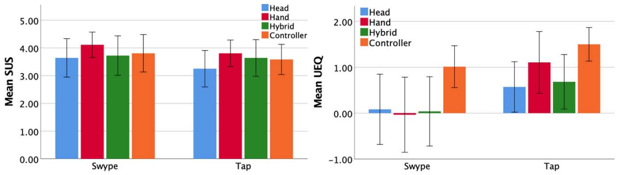

Figure 8: Mean immersion score from SUS questionnaire (a; left). Mean user experience score from UEQ (b; right). Error bars indicate ± 2 standard errors.

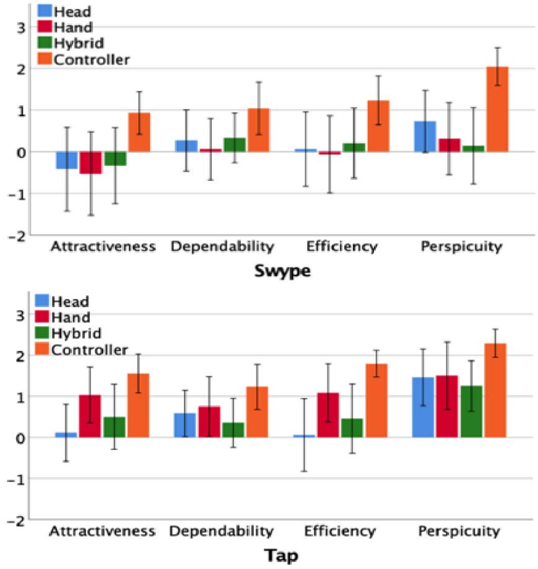

Figure 9: Mean UEQ subscales for each pointing method for Swype (a; top) and in Tap (b; bottom). Error bars indicate ± 2 standard errors.

For Tap, the test yielded a significant effect of Pointing Methods ($F_{2.137,23.503} = 39.971, p < .001$). Pairwise comparison revealed significant differences between Controller - Head, Controller - Hybrid, Controller - Hand (all $p < .001$). For Swype, the test yielded a significant effect of Pointing Methods ($F_{1.974,21.719} = 89.375, p < .001$). Post-hoc pairwise comparison revealed significant differences between Controller - Head ($p < .001$), Controller - Hybrid ($p < .001$), Controller - Hand ($p < .001$), and Head - Hybrid ($p < .01$).

3.4.2 User Preference

SUS. The SUS counts for Hand+Swype (M = 1.08, SD = 1.62) were the highest but the lowest for Controller+Tap (M = 0.17, SD = 0.39). Figure 8a shows that the mean immersion score from SUS questionnaire for Hand+Swype (M = 4.11, SD = 0.80) was the highest and Head+Tap (M = 3.25, SD = 1.14) the lowest. There was no significant difference for immersion between the Pointing Methods ($F_{3,66} = 3.199, p = .029$), Pointing Methods × Input Mechanisms ($F_{3,66} = .308, p = .820$), and Input Mechanisms ($F_{1,22} = .419, p = .524$).

UEQ. The scales for UEQ was adjusted between -3 (very bad) to 3 (excellent). For the average score, ANOVA tests showed a significant effect of Pointing Methods ($F_{3,66} = 9.295, p < .001$), but insignificant for Pointing Methods × Input Mechanisms ($F_{3,66} = 1.183, p = .322$). There was no significant effect of Input Mechanisms ($F_{1,22} = 3.306, p = .083$) where the average experience score for Tap was 0.965 (SD = 1.01) and for Swype 0.275 (SD = 1.27). Post-hoc pairwise comparisons revealed significant differences between Head - Controller ($p < .001$) and Hybrid - Controller ($p < .01$). Figure 8b shows the details of the mean UEQ for all methods.

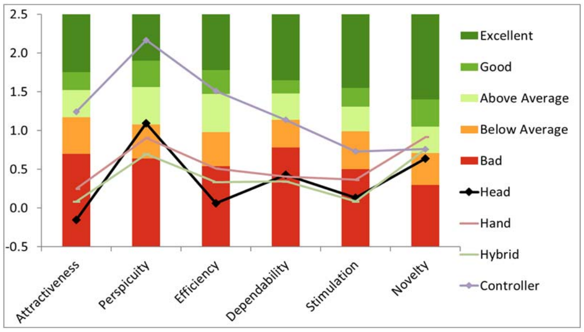

Figure 10: User Experience Questionnaire (UEQ) ratings of our tested pointing methods (Head, Hand, Hybrid, and Controller) with respect to comparison benchmarks

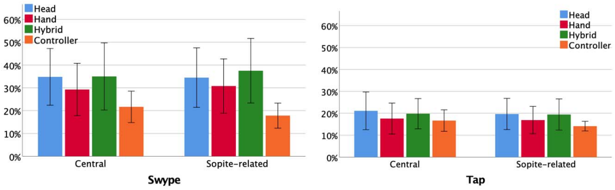

Figure 11: MSAQ subscales for each Pointing Method in Swype (a; left) and in Tap (b; right). Peripheral is not shown as no significant difference was found. Error bars indicate ± 2 standard errors.

Regarding each UEQ subscale (see Fig. 9), ANOVA tests yielded a significant effect of Pointing Method, Input Mechanisms, or Pointing Methods × Input Mechanisms on attractiveness, perspicuity, efficiency, and dependability. However, no significant effect was found for novelty and stimulation. Table 3 shows detailed results of the ANOVA tests. As can be seen from the Fig. 10, the controller was rated above average to excellent when compared to the benchmark scores while the other three pointing methods were rated between bad and above average.

Motion Sickness. Regarding overall motion sickness, Controller+Tap was rated the best (M = 14.53%, SD = 4.92%) and Hybird+Swype (M = 30.09%, SD = 18.54%) the worst. ANOVA tests yielded significant differences between Pointing Methods ($F_{2.694,59.262} = 5.662, p < .01$); however, no significant effect was found for Pointing Methods × Input Mechanisms ($F_{2.694,59.262} = 1.942, p = .138$) and Input Mechanisms ($F_{1,22} = 4.435, p = .047$). Pairwise comparisons did not reveal any significant differences.

Regarding the MSAQ subscales (gastrointestinal, central, peripheral, and sopite-related), there was a significant effect of Pointing Methods ($F_{3,66} = 4.979, p < .01$) on central. However, post-hoc pairwise comparison yielded no significant difference. In terms of sopite-related motion sickness, the ANOVA test yielded significant differences between Pointing Methods ($F_{3,66} = 8.406, p < .001$), but not between Pointing Methods × Input Mechanisms ($F_{3,66} = .808, p = .067$). Post-hoc pairwise comparison showed a significant difference between Head - Controller and Hybrid - Controller (all $p < .01$). No other significant effects were found. Figure 11 shows MSAQ subscales scores.

NASA-TLX Workload. For overall task workload, Controller+Tap was rated the best (M = 33.92, SD = 19.44) and Hybrid+Swype (M = 75.19, SD = 12.84) the worst. An ANOVA test showed significant differences for Pointing Methods ($F_{3,66} = 26.063, p < .001$) on overall workload, but not for Pointing Methods × Input Mechanisms ($F_{3,66} = 3.990, p = .011$) and Input Mechanisms ($F_{1,22} = 5.724, p = .026$). Post-hoc pairwise comparisons revealed significant differences between Head - Controller, Hand - Controller, Hybrid - Controller (all $p < .01$). Regarding each workload subscale, ANOVA tests yielded at least one significant effect for Pointing Methods on all workload subscales except for performance. Details of results of the ANOVA tests can be seen in Table 4 and of the workload subscales in Fig. 12.

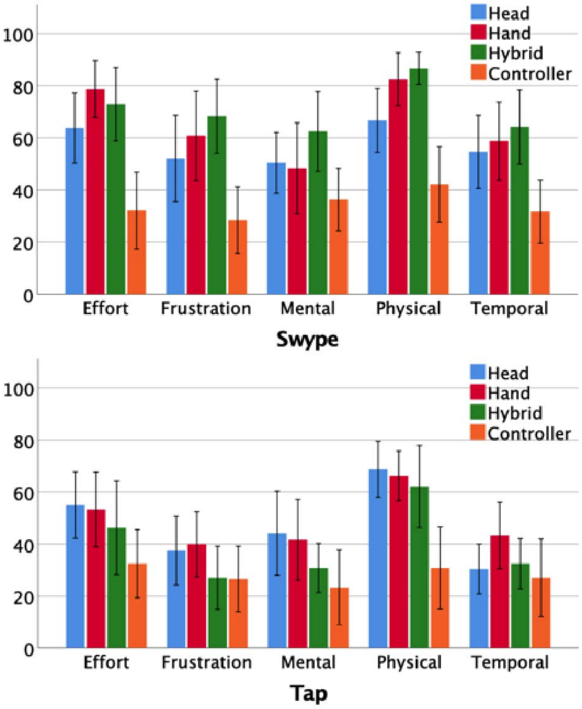

Figure 12: Workload subscales for each Pointing Method using Swype (a; top) and Tap (b; bottom). Performance is non-significant and not shown for better clarity. Error bars indicate ± 2 standard errors.

4 DISCUSSION

In this section, we first discuss task performance of the combination of each Pointing Method (Head, Hand, Hybrid, and Controller) and Input Mechanism (Swype and Tap), and then user subjective feedback for each combination.

4.1 Task Performance

Controller+Tap achieved an average of 14.6 WPM which is comparable to results in other pointing method + tap approaches from research in VR and non-VR domains [29,33,35]. Controller+Swype achieved an average of 13.68 WPM which is also comparable to some results in VR (e.g., in [42] their participants were able to achieve 15.75 WPM). Our results also indicate that Controller outperformed all the other device-free methods. However, when compared to a physical keyboard, which has been shown to be able to support fast text entry of around 45 to 67 WPM [20], Controller-based input seems still not fast enough for heavy text entry activities. It may not be necessarily an issue with any pointing method in particular but that AR in general may not support long periods of intensive text entry. For short text entry tasks like sending short messages via social media chat applications, a technique based on Controller+Swype could work well.

Head+Tap has led to an average of 5.62 WPM using the outside-inside approach but this is only the half of input speed of Head Pointing using a button to make selections (about 10 WPM) [35,42]

Table 4: ANOVA test results for NASA-TLX workload subscales. Significant results where $p < .01$ are shown in green and $p < .001$ in dark green. Non-significant results are omitted for clarity.

	Pointing Methods	Input Mechanisms	Post-hoc Pointing Methods
Mental	$F_{3,66} = 4.813, p < .01$	$F_{1,22} = 3.571, p = .072$	N/A
Physical	$F_{3,66} = 22.021, p < .001$	$F_{1,22} = 5.081, p = .034$	Head - Controller, Hand - Controller, Hybrid - Controller (all $p < .001$)
Temporal	$F_{3,66} = 6.975, p < .01$	$F_{1,22} = 8.175, p < .01$	Hand-Controller ($p < .01$)
Effort	$F_{3,66} = 13.045, p < .001$	$F_{1,22} = 4.867, p < .038$	Head - Controller, Hybrid - Controller (both $p < .01$), Hand - Controller ($p < .001$)
Frustration	$F_{3,66} = 6.004, p < .01$	$F_{1,22} = 8.537, p < .01$	Hand - Controller ($p < .01$)

and is also slower than the dwell-based Head Pointing technique (around 8 WPM) [42]. Unlike Hand and Hybrid, both of which can use hand gestures for letter/word selection, there are currently no optimal methods, except dwell, for purely Head Pointing-based approaches for text entry with a QWERTY keyboard layout. If users have to use Head Pointing-based approaches, an alternative approach that exists in the literature is to use a circular layout like a technique called RingText [41] which has been shown to be faster than dwell QWERTY.

We have observed that text entry performance (both speed and accuracy) for Head and Hybrid are affected by the hardware (e.g., tracking cameras and feasible tracked area), software (e.g., gesture detection algorithms), and users' physical capabilities and predispositions (e.g., how long and how stable they can hold their hand in mid-air). In the context of AR, the area that is tracked by the cameras tend to be limited and because of this the users must lift their hands in the mid-air, which may further cause hand tremor and arm fatigue quickly, making it challenging for many users. The detection algorithm provided by the Meta SDK seems to have issues. We have discovered that, when the users move their hand out of the tracking area accidentally or intentionally, the algorithm sometimes thinks that their hands are performing a palm closing gesture—i.e., a false positive recognition, and assumes a selection is made while in fact, the users are not doing anything. Because of this, during the experiment, we had to remind users to keep their hand within the tracking area.

As for Input Mechanisms, the experimental results suggest that for users who are new to Swype and Tap, the Swype technique have the same text entry speed as Tap and cause lower NCER than Tap. If users prefer lower errors in the transcribed text, they should use Swype instead of Tap.

4.2 User Preference

In the following discussion, we discuss each Pointing Method and Input Mechanism based on the subjective feedback and our observations from the experiment.

4.2.1 Workload

Controller outperformed all the other methods for Physical and Effort workload and exceeded Hand for Temporal and Frustration. As such, a Controller-type of input seems to be a good first option if a lower workload is important for users. Our observations also show that our participants complained that Hand and Hybrid were too tiring because of the need to hold their hands in mid-air in a consistent and stable basis. Due to the limitations of the Meta 2 headset's tracking area, the users cannot place their hands in a more relaxing pose. It is worth pointing that this issue is not just confined to the Meta 2 but it is a widely report issues for AR devices. Although Head did not have this problem, participants complained about minor neck pain and fatigue. One solution could be to use a device with an eye-tracking device installed (i.e., gaze input [30]), if the cost is not an issue and the eye tracker can provide accurate and stable performance. Thus, when a controller is not around, users could consider a Head approach when hand fatigue is a big concern. They should consider a Hand approach when arm fatigue is less of an issue.

Swype techniques resulted in a significantly higher temporal and frustration workload than Tap. Surprisingly, Swype and Tap have the same level of mental workload even though Swype requires users to remember and type all letters in one continuous Swype action to complete the words. It is worth noting that although our participants were not native alphabet users they were still able to mentally keep track of the words that they needed to type using Swype with relative proficiency, but this had come with higher frustration and temporal workload, which may not be the case with English native speakers. In general, if the workload is a critical factor of the text entry technique, a Swype-style text approach should not be considered due to its high workload demand in both temporal and frustration workload.

4.2.2 Motion Sickness

Results indicated no differences for the overall sickness among the tested techniques. For each subscale from motion sickness assessment questionnaire [11], the Controller approach was found to be less annoying, drowsing and tiring than Head and Hybrid techniques because it did not need our participants to use head rotations. This means that a ray-casting enabled controller should be preferred if available. Additionally, users should consider a Hand approach when the controller is not around.

For Input Mechanisms, our results indicate that Tap causes the same level of motion sickness as Swype. The selection of which Input Mechanism to apply should consider other aspects (e.g., workload) as they both have no effect on motion sickness.

4.2.3 Immersion

There were no significant differences between the difference combinations of pointing methods and input mechanisms for immersion, which indicates that text entry in AR has no significant impact on immersion. Overall, users should consider other factors (e.g., workload) to decide which technique to use.

4.2.4 User Experience

For the user experience subscales, Controller provided a significantly better user experience in efficiency and perspicuity than the other methods. It also gave better dependability than Hybrid and received higher ratings in attractiveness than Hybrid and Hand. When we compare these pointing approaches with the benchmark scores [32], only Controller is found to have received an above average to excellent rating while Head, Hand, and Hybrid are rated bad to below average. For the input mechanisms, we found that Tap and Swype have no significant difference on user experience.

In summary, the Controller offers the best user experience and as such, if a ray-casting enabled controller is available, it should be used as a first choice. Otherwise, users should consider other user experience measurements such as workload to decide which alternative pointing methods to use.

4.3 Recommendations for Text Entry in AR HMDs

The recommendations derived from our experiment can be divided into two groups based on their goals:

Performance. Based on the results, we suggest that users should use a ray-casting enabled handheld device since it can lead to a good text entry performance and it is capable of other tasks, like manipulating virtual objects [42,43]. Device-free methods should be considered in addition to speech recognition, if available, when device-free is the only option. On the other hand, if the environment is noisy and users are in a public space, which can potentially bring privacy concerns [40], we suggest using one of the device-free approaches based on user experience. Of the two input mechanisms, Swype should be considered first since it has a higher text entry rate and a lower not corrected error rate than Tap.

User Experience. We suggest that a handheld device should be the preferred option because it has low workload and motion sickness but provides a better user experience. However, if no such devices are around, the following can be considered. If users have difficulty holding their hands constantly and consistently in mid-air, Head-based pointing can be considered as an alternative. Hybrid can be used if arm and neck fatigue is not a concern and there is enough space for users to lift and hold the arms mid-air. If users' neck fatigue is a concern and users have ample space for hand interaction, the Hand approach could be chosen instead. This is also because a natural hand interaction allows users to perform tasks in both the real and virtual environment at the same time [5]. Of the two input mechanisms, Tap should be chosen since it generates lower workload (for both temporal and frustration).

4.4 Limitations and Future Work

This research has some limitations. The experiment was tested with a Meta 2 AR HMD. We chose it because it had one of widest field-of-view and, like other AR devices, it has some issues in tracking hand motions and gestures. We used 3 countermeasures to minimize issues that this could have caused: (1) We chose one of the most simple gestures (closing palm) which the Meta 2 provided and of which it had a reliable tracking performance; (2) To avoid potential environmental noise factors that may affect tracking performance, we did tests to ensure the environment would not cause any tracking issues; and (3) We allowed users to familiarize themselves with the device and techniques via warm-up practices. Given this, the AR device chosen in our study is still suitable for our purposes and the results we obtained are still quite relevant to AR systems. In the future, when AR devices have improved tracking performance, it will be useful to explore other combinations of pointing and selection methods for entering the text that is accurate and fast.

We observed that with the number of phrases that our participants had to type, some of them felt that their hand and arm got tired, especially for the Hand and Hybrid approaches. Future research can explore possible ways to minimize arm/hand fatigue for these two types of approaches. Similarly, our experiment involved 12 participants in each group (24 in total), which according to Caine [6] is one of the most common sample sizes within HCI research. Given our sample size, we used the alpha value of 0.01 to ensure that any replication could likely achieve similar results [2]. In the future, it will be useful to evaluate if performance and user experience can improve with larger sample size and longer experimental sessions, for example, 1-2 sessions over consecutive 4-5 days like PizzaText [43], RingText [41] in VR scenarios and WrisText [15] in smartwatch scenarios.

Additionally, our evaluation experiment was conducted in a lab environment where the background is somewhat, but not fully, controlled to be clean and easy for the front camera to track the hand motions and gestures. Future work can consider experimenting with more realistic environments, e.g., in a park or a shopping mall with people walking in front of the camera. This future research can be informed by the results of this current experiment.

Finally, as mentioned in the discussion section, the selected pointing methods and AR devices in general may not be suitable for long text entry sessions and heavy text editing of documents. Although AR devices are usually meant for short text entry sessions (like for sending short messages), it is worthwhile to explore and develop new techniques that will support text entry activities that are more involved and last longer. For instance, easily and widely accessible devices like smartphones, which have been reported to support users to type 50 WPM when they are sitting [7] and about 30 WPM when they are walking [13], can be part of this exploration. Also, voice input techniques, such as SilentVoice [9] which can mitigate privacy issues and work well in noisy environments, are also valuable and can be useful for some text entry activities. Further research is needed because both smartphones or SilectVoice have their inherent technical and usability issues and, if we are to develop new techniques that linked them to an AR system, these issues need to be overcome.

5 SUMMARY AND CONCLUSION

In this work, we empirically and systematically investigated the combination of four pointing methods (Head, Hand, Hybrid, and Controller) and two input/selection mechanisms (Swype and Tap) that can be used for text entry in augmented reality (AR) head-mounted displays (HMDs). We run an experiment with the 8 techniques that resulted from their combinations to assess their relative performance and user preference. In general, the results show that the best pointing method is a ray-casting enabled handheld device, but its use is dependent on specific criteria and limitations (e.g., ray-casting enabled controller is not always available for AR systems, or users cannot hold it in a stable basis). Future AR systems may be commonly used for both indoor and outdoor scenario but a ray-casting enabled controller may not be ideal for outdoor situations. Therefore, a device-free efficient text entry method is still a more practical and cost-efficient solution because it only requires the HMD to be able to track a user's hand or head motions. On the other hand, user preference such as workload and user experience must be considered also. Between the two selection mechanisms that we explored, Swype and Tap, our results show that Swype is as fast as Tap for users who are new to Swype. But Swype brings increased workload (i.e., temporal and frustration). For lighter workload during text entry activities, users can use Tap. Our research is a first to explore the combination of most common pointing methods and selection mechanisms and can provide strong foundations for future research in text entry for augmented reality systems.

ACKNOWLEDGMENTS

We thank all the participants for the time and the reviewers for their comments and suggestions that have helped improve our paper. This research was supported by AI University Research Centre (AI-URC) at Xian Jiaotong-Liverpool University (XJTLU), XJTLU Key Program Special Fund (KSF-A-03 and KSF-02), and XJTLU Research Development Fund.

REFERENCES

[1] The most frequent 10 000 words of English and Example Sentences - The list of most commonly used words. `http://www.use-in-a-sentence.com/english-words/10000-words/the-most-frequent-10000-words-of-english.html`. Accessed: 2018-04-01.

[2] A. A. Aarts, J. E. Anderson, C. Anderson, P. Attridge, A. Attwood, J. Axt, M. Babel, . Bahnk, E. Baranski, M. Barnett-Cowan, E. Bartmess, J. Beer, R. Bell, H. Bentley, L. Beyan, G. Binion, D. Borsboom, A. Bosch, F. A. Bosco, and M. Penuliar. Estimating the reproducibility of psychological science. *Science*, 349, 08 2015. doi: 10.1126/science.aac4716

[3] F. Argelaguet and C. Andujar. A survey of 3d object selection techniques for virtual environments. *Computers & Graphics*, 37(3):121–136, May 2013. doi: 10.1016/j.cag.2012.12.003

[4] S. Azenkot and S. Zhai. Touch behavior with different postures on soft smartphone keyboards. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, pp. 251–260. ACM, 2012.

[5] R. T. Azuma. A Survey of Augmented Reality. *Presence: Teleoperators & Virtual Environments*, 6(4):355–385, 1997.

[6] K. Caine. Local Standards for Sample Size at CHI. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*, pp. 981–992. ACM Press, Santa Clara, California, USA, 2016. doi: 10.1145/2858036.2858498

[7] J. Cuaresma and I. S. MacKenzie. A study of variations of qwerty soft keyboards for mobile phones. In *Proceedings of the International Conference on Multimedia and Human-Computer Interaction-MHCI*, pp. 126–1, 2013.

[8] N. A. ElSayed, B. H. Thomas, K. Marriott, J. Piantadosi, and R. T. Smith. Situated Analytics: Demonstrating immersive analytical tools with Augmented Reality. *Journal of Visual Languages & Computing*, 36:13–23, Oct. 2016. doi: 10.1016/j.jvlc.2016.07.006

[9] M. Fukumoto. Silentvoice: Unnoticeable voice input by ingressive speech. pp. 237–246, 10 2018. doi: 10.1145/3242587.3242603

[10] W. Garbe. SymSpell: 1 million times faster through Symmetric Delete spelling correction algorithm, Mar. 2018. original-date: 2014-03-25T11:01:35Z.

[11] P. J. Gianaros, E. R. Muth, J. T. Mordkoff, M. E. Levine, and R. M. Stern. A Questionnaire for the Assessment of the Multiple Dimensions of Motion Sickness. *Aviation, space, and environmental medicine*, 72(2):115–119, Feb. 2001.

[12] Y. Gizatdinova, O. pakov, and V. Surakka. Comparison of video-based pointing and selection techniques for hands-free text entry. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pp. 132–139. ACM, 2012.

[13] M. Goel, L. Findlater, and J. Wobbrock. Walktype: Using accelerometer data to accommodate situational impairments in mobile touch screen text entry. In *Proceedings of the 2012 CHI Conference on Human Factors in Computing Systems - CHI '12*. ACM Press, 05 2012. doi: 10.1145/2207676.2208662

[14] D. Goncharov. Implement the same algorithm "swype keyboard" for .NET and Unity.: Neargye/SwipeType, June 2018. original-date: 2018-03-14T15:34:52Z.

[15] J. Gong, Z. Xu, Q. Guo, T. Seyed, X. Anthony' Chen, X. Bi, and X.-D. Yang. Wristext: One-handed text entry on smartwatch using wrist gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pp. 1–14. ACM Press, 04 2018. doi: 10.1145/3173574.3173755

[16] G. Gonzlez, J. P. Molina, A. S. Garca, D. Martnez, and P. Gonzlez. Evaluation of Text Input Techniques in Immersive Virtual Environments. In J. A. Macas, A. Granollers Saltiveri, and P. M. Latorre, eds., *New Trends on Human-Computer Interaction*, pp. 1–10. Springer London, London, 2009. doi: 10.1007/978-1-84882-352-5_11

[17] J. Grubert, L. Witzani, E. Ofek, M. Pahud, M. Kranz, and P. O. Kristensson. Effects of hand representations for typing in virtual reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 151–158, March 2018. doi: 10.1109/VR.2018.8446250

[18] J. Grubert, L. Witzani, E. Ofek, M. Pahud, M. Kranz, and P. O. Kristensson. Text Entry in Immersive Head-Mounted Display-Based Virtual Reality Using Standard Keyboards. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 159–166, Mar. 2018. doi: 10.1109/VR.2018.8446059

[19] S. G. Hart. NASA Task load Index (TLX). Volume 1.0; Paper and pencil package. 1986.

[20] P. Knierim, V. Schwind, A. M. Feit, F. Nieuwenhuizen, and N. Henze. Physical Keyboards in Virtual Reality: Analysis of Typing Performance and Effects of Avatar Hands. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pp. 1–9. ACM Press, Montreal QC, Canada, 2018. doi: 10.1145/3173574.3173919

[21] E. Kruijff, J. E. Swan, and S. Feiner. Perceptual issues in augmented reality revisited. In *2010 IEEE International Symposium on Mixed and Augmented Reality*, pp. 3–12, Oct 2010. doi: 10.1109/ISMAR.2010.5643530

[22] A. Kurauchi, W. Feng, A. Joshi, C. Morimoto, and M. Betke. EyeSwipe: Dwell-free Text Entry Using Gaze Paths. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*, pp. 1952–1956. ACM Press, Santa Clara, California, USA, 2016. doi: 10.1145/2858036.2858335

[23] B. Laugwitz, T. Held, and M. Schrepp. Construction and Evaluation of a User Experience Questionnaire. In A. Holzinger, ed., *HCI and Usability for Education and Work*, vol. 5298, pp. 63–76. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. doi: 10.1007/978-3-540-89350-9_6

[24] J.-W. Lin, P.-H. Han, J.-Y. Lee, Y.-S. Chen, T.-W. Chang, K.-W. Chen, and Y.-P. Hung. Visualizing the Keyboard in Virtual Reality for Enhancing Immersive Experience. In *ACM SIGGRAPH 2017 Posters*, SIGGRAPH '17, pp. 35:1–35:2. ACM, New York, NY, USA, 2017. doi: 10.1145/3102163.3102175

[25] A. Liverani, G. Amati, and G. Caligiana. A CAD-augmented Reality Integrated Environment for Assembly Sequence Check and Interactive Validation. *Concurrent Engineering*, 12(1):67–77, Mar. 2004. doi: 10.1177/1063293X04042469

[26] K. Lyons, T. Starner, D. Plaisted, J. Fusia, A. Lyons, A. Drew, and E. W. Looney. Twiddler typing: one-handed chording text entry for mobile phones. In *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*, pp. 671–678. ACM Press, Vienna, Austria, 2004. doi: 10.1145/985692.985777

[27] I. S. MacKenzie and R. W. Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI'03 extended abstracts on Human factors in computing systems*, pp. 754–755. ACM, 2003.

[28] S. Malik, C. McDonald, and G. Roth. Hand tracking for interactive pattern-based augmented reality. In *Proceedings. International Symposium on Mixed and Augmented Reality*, pp. 117–126. IEEE Comput. Soc, Darmstadt, Germany, 2002. doi: 10.1109/ISMAR.2002.1115080

[29] A. Markussen, M. R. Jakobsen, and K. Hornbk. Selection-Based Mid-Air Text Entry on Large Displays. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, P. Kotz, G. Marsden, G. Lindgaard, J. Wesson, and M. Winckler, eds., *Human-Computer Interaction INTERACT 2013*, vol. 8117, pp. 401–418. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. doi: 10.1007/978-3-642-40483-2_28

[30] M. E. Mott, S. Williams, J. O. Wobbrock, and M. R. Morris. Improving Dwell-Based Gaze Typing with Dynamic, Cascading Dwell Times. pp. 2558–2570. ACM Press, 2017. doi: 10.1145/3025453.3025517

[31] J. Orlosky, K. Kiyokawa, and H. Takemura. Dynamic text management for see-through wearable and heads-up display systems. pp. 363–370, 03 2013. doi: 10.1145/2449396.2449443

[32] M. Schrepp, A. Hinderks, and J. Thomaschewski. Construction of a Benchmark for the User Experience Questionnaire (UEQ). *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(4):40, 2017. doi: 10.9781/ijimai.2017.445

[33] G. Shoemaker, L. Findlater, J. Q. Dawson, and K. S. Booth. Mid-air text input techniques for very large wall displays. In *Proceedings of Graphics interface 2009*, pp. 231–238. Canadian Information Processing Society, 2009.

[34] R. W. Soukoreff and I. S. MacKenzie. Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pp. 113–120. ACM, New York, NY, USA, 2003. doi: 10.1145/642611.642632

[35] M. Speicher, A. M. Feit, P. Ziegler, and A. Krger. Selection-based Text Entry in Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*, pp. 1–13. ACM Press, Montreal QC, Canada, 2018. doi: 10.1145/3173574.3174221

[36] S. Sridhar, A. M. Feit, C. Theobalt, and A. Oulasvirta. 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*, pp. 3643–3652. ACM Press, Seoul, Republic of Korea, 2015. doi: 10.1145/2702123.2702136

[37] B. H. Thomas and W. Piekarski. Glove Based User Interaction Techniques for Augmented Reality in an Outdoor Environment. *Virtual Reality*, 6(3):167–180, Oct. 2002. doi: 10.1007/s100550200017

[38] D. van Krevelen and R. Poelman. A Survey of Augmented Reality Technologies, Applications and Limitations. *The International Journal of Virtual Reality*, 9(2):1–20, June 2010.

[39] A. D. Wilson and M. Agrawala. Text entry using a dual joystick game

controller. In *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*, p. 475. ACM Press, Montréal, Québec, Canada, 2006. doi: 10.1145/1124772.1124844

[40] J. O. Wobbrock, D. H. Chau, and B. A. Myers. An alternative to push, press, and tap-tap-tap: gesturing on an isometric joystick for mobile phone text entry. In *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07*, p. 667. ACM Press, San Jose, California, USA, 2007. doi: 10.1145/1240624.1240728

[41] W. Xu, H. Liang, Y. Zhao, T. Zhang, D. Yu, D. Monteiro, and Y. Yue. RingText: Dwell-free and hands-free Text Entry for Mobile Head-Mounted Displays using Head Motions. *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2019. doi: 10.1109/TVCG.2019.2898736

[42] C. Yu, Y. Gu, Z. Yang, X. Yi, H. Luo, and Y. Shi. 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 - CHI '17*, pp. 4479–4488. ACM Press, Denver, Colorado, USA, 2017. doi: 10.1145/3025453.3025964

[43] D. Yu, K. Fan, H. Zhang, D. Monteiro, W. Xu, and H. Liang. PizzaText: Text Entry for Virtual Reality Systems Using Dual Thumbsticks. *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2927–2935, Nov. 2018. doi: 10.1109/TVCG.2018.2868581