Text Entry in Virtual Reality: A Comprehensive Review of the Literature

Tafadzwa Joseph Dube(✉) and Ahmed Sabbir Arif

Human-Computer Interaction Group,
University of California, Merced, CA 95343, USA
{tdube, asarif}@ucmerced.edu

Abstract. The availability of consumer-ready Virtual Reality (VR) Head-Mounted Displays (HMDs) has resulted in a surge in VR applications. It has also prompted the design and development of numerous text entry techniques for the paradigm. However, it is difficult to understand the mechanism of these techniques and extract meaningful average performance data from this body of work since they were evaluated in diverse experiment conditions and report different performance metrics. To remedy this, this paper classifies the existing text entry techniques for VR based on their input mechanism and discusses their strengths, limitations, and performance.

Keywords: Virtual Reality · Text entry · Review · Performance · Input

1 Introduction

Virtual Reality (VR) is rapidly growing in popularity due to the availability of affordable consumer-ready products [1]. Although initially branded as a technology for entertainment and simulation, nowadays its being used for office work [2, 3], collaboration [4], and training and education [5]. These newfound applications have rendered a need for efficient and effective text entry techniques for VR as inputting text is an essential part of these experiences. Many recent works have attempted to meet this need by developing novel, as well as customizing the existing text entry techniques for VR. However, it is often difficult to comprehend the mechanism of these techniques and extract meaningful average performance data from this body of work as they were evaluated in different experiment conditions and report different performance metrics. This makes it difficult for the researchers to use and apply these findings, causing re-exploration of design philosophies, and as a result slowing down the overall development process. To address this, this paper reviews the existing text entry techniques for VR. It categorizes these techniques based on their input mechanism and discusses their strengths, limitations, and performance. It also provides design recommendations for researchers to facilitate the development of more user-friendly and effective text entry techniques. This work does not include speech recognition.

M. Kurosu (Ed.): HCII 2019, LNCS 11567, pp. 419–437, 2019.
https://doi.org/10.1007/978-3-030-22643-5_33

2 Physical Techniques

Adapting the standard physical Qwerty keyboard for VR is difficult for several reasons. Numerous studies have established that both novice and expert typists look down at the keyboard to verify their hand position [6, 7]. Since users wearing a Head-Mounted Display (HMD) cannot see their hands (Fig. 1), it is almost impossible for them to input text as fast and accurate as in the real world. The size of the keyboards and the need for a supporting surface also make physical Qwerty harder to use in scenarios where users are required to move around. Although some have proposed miniature Qwerty layouts, the smaller key size makes it impossible to touch-type, which further affects entry speed and accuracy [8]. This section reviews all techniques that use a physical keyboard or a keypad in VR. Table 1 presents the entry speed and error rates [9] of these techniques from the literature.

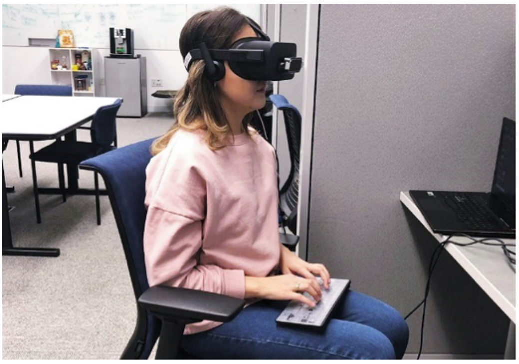

Fig. 1. A user entering text in VR using a physical Qwerty keyboard.

2.1 Physical Qwerty

Many have used a physical Qwerty or its variants, particularly Qwertz and Azerty, in VR to bank on their widespread use and familiarity [10–12]. Most of these techniques track the keyboard and hands with external depth cameras to display their virtual representations in VR [2, 10, 13, 14] to enable users to enter text with a physical keyboard through its animated representation. In an evaluation, this approach yielded 34.0 wpm and 12% error rate [13].

Bovet et al. [14] used the Logitech BRIDGE SDK [15] with an HTC Vive Pro HMD [16] and a Logitech G810 Orion Spectrum keyboard to display an animated representation of the keyboard and hands in VR. In a user study, this method reached an average entry speed of 44.4 wpm from an initial speed of 34.5 wpm. It also yielded negligible mental and physical demand scores. Grubert et al. [2] argued that this method can retain at least 50% of users' desktop typing skills. This method's effectiveness, however, is dependent on the reliability of its tracking sensors.

Kim and Kim [17] tracked the keyboard and hand positions to display a visual representation of the keyboard and hands. Unlike the previous approach, they did not track fingers, instead displayed finger positions based on the last keypress. That is, they calibrated finger positions based on the assumption that specific fingers were used to press specific keys, which is a cheaper, but a less accurate approach compared to using depth cameras. In an evaluation, users retained at least 60% of their desktop typing speed and 80% of accuracy with this method.

A different method embedded a live video stream of the real world onto the virtual world to afford the use of a physical Qwerty in VR [11, 12]. It displayed the full and a partial view of the keyboard and hands. In a user study, the full view yielded 36.7 wpm and 10.4% error rate [12]. Partial view that displayed only the keyboard and the hands yielded comparable results. McGill et al. [12] also proposed a partial blending option that displays parts of the keyboard but did not evaluate its performance. Although this method demonstrates competitive speed and accuracy, it breaks the immersion by displaying the real world. It also requires the use of additional cameras, increasing the cost of development and installation. To remedy this, Walker et al. [18] proposed a software solution for predicting finger positions. They argued that when expert users are provided with adequate visual feedback, they can use their sense of proprioception to correctly (re)position their hands on the keyboard for the next key. They used a decoder [19] for auto-correction. In a user study, this approach reached 43.7 wpm and 92.6% accuracy rate. Although the performance of this approach is promising, its reliance on a decoder can potentially frustrate users. Its dependence on the sense of proprioception can also increase the cognitive load.

It is clear from this section that physical Qwerty remains an essential method for text entry in immersive environments. Researchers have proposed the use of various sensors, cameras, and decoders to facilitate reasonably fast and accurate text entry in VR with physical Qwerty, which may not be feasible in all scenarios and for low-income groups.

2.2 Mobile Keypads

Some immersive systems allow users to move around within a limited space. Since physical Qwerty restricts this ability, researchers have proposed using mobile keypads.

Bowman et al. [20] and González et al. [21] used Twiddler in VR. Twiddler is a 12-key chorded keypad, with which users enter characters by pressing either one or a combination of keys simultaneously (called a chord). This approach was evaluated in a user study, where a virtual layout of the keypad was displayed in VR. In the study, Twiddler yielded 3.0 wpm and 82% accuracy rate. Although Twiddler enables users to be mobile, it is substantially slower than the other techniques. Besides, it requires extensive training to learn the chords, which may discourage many to try it [6].

González et al. [21] also evaluated a 9-key mobile keypad that has three characters on each key. They embossed the keys to provide users with haptic feedback. To enter characters with this keypad, users press the key containing the intended character once or multiple times, respective to the position of the character on the key, just like Multi-tap [6]. For instance, to enter the letter “c”, users press the “2” key containing the letters “a”, “b”, and “c” three times. This method yielded 12.1 wpm and 95% accuracy rate. Although relatively slow, this method could be useful for short-term text entry, such as while entering a password or a search keyword.

Table 1. Physical text entry techniques with their entry speed in words per minute (wpm) and error rate (%). "Char" signifies "character". Sensors are not considered as auxiliary devices.

Method	Short description	Input unit	Auxiliary devices	Bimanual	Hand tracking	Participant	Session	Entry speed (wpm)	Error rate (%)
Qwerty [12]	_Displays full* and partial† view of the keyboard and the hands_	Char	Keyboard Camera	✓	✓	16	1	36.6* 38.5†	10.4* 9.2†
Qwerty [18]	_Visual feedback on keypress without* & with† auto-correction_	Char	Keyboard	✓	✗	24	2	43.7	8.4* 2.6†
Qwerty [13]	_Displays an animated view of the keyboard without* & with† hands_	Char	Keyboard	✓	✓	13	1	34.0* 31.2†	12.0* 13.4†
Qwerty [17]	_Calibrates virtual finger position based on the last keypress_	Char	Keyboard	✓	✓	✗	1	✗	✗
Qwerty [14]	_Displays an animated view of the keyboard and hands_	Char	Keyboard	✓	✓	12	1	44.4	0.4
Qwerty [22]	_Tests four feedback conditions: none*, animated keyboard†, animated keyboard & hands‡, & video inlay of keyboard & hands§_	Char	Keyboard Keyboard Camera	✓	✓	16	1	28.1* 24.3† 27.4‡ 27.8§	0.3* 0.2† 0.2‡ 0.2§
Qwerty [10]	_Tests four feedback conditions: none*, animated fingertips†, animated abstract hands ‡, & animated realistic hands§_	Char	Keyboard Keyboard Camera	✓	✓	16	1	31.8* 32.2† 38.8‡ 37.5§	14.0* 7.5† 7.7‡ 7.6§
	Same four conditions evaluated with experienced typists whose entry speed is over 53.3 wpm		Keyboard Keyboard Camera			16	1	61.8* 68.5† 67.0‡ 66.6§	7.3* 4.4† 5.1‡ 5.0§
Qwertz [11]	_Tests four feedback conditions: keypress only*, animated fingertips†, animated hands‡, & video inlay of keyboard & hands§_	Char	Keyboard Camera	✓	✓	24	1	26.1* 36.4† 34.4‡ 38.7§	15.2* 6.3† 11.5‡ 5.1§
Qwertz [2]	_Users look straight* or down† to see the keyboard_	Char	Keyboard Camera	✓	✓	24	1	25.3* 26.3†	2.4* 2.1†
Chorded [20, 21]	_Displays an animated view of the chorded keyboard_	Char	Twiddler	✗	✗	10	1	3.0 [21]	17.0
Keypad [21]	_Uses a custom embossed keypad to provides haptic feedback_	Char	X-keys [23]	✗	✗	10	1	12.1	4.0

2.3 Discussion

One can see in Table 1, Knierim et al. [10], Grubert et al. [11], Hoppe et al. [13], Bovet et al. [14], and Lin et al. [22] all investigated physical Qwerty with animated keyboard and hands as visual feedback, yet reported diverse results, summarized in Table 2. Bovet et al.'s [14] study yielded the highest entry speed (roughly16% faster than the next fastest result). The fact that the baseline condition of this study also yielded a relatively high entry speed suggests that this is likely due to a more experienced sample. The fact that this study used a sophisticated tracking apparatus optimized for text entry in VR may have contributed towards this as well. This presumably impacted the accuracy rate of the method too since Bovet et al. [14] and Lin et al. [22] reported substantially lower error rates than the others (0.4% vs. > 7.6%).

Table 2. Studies investigating physical Qwerty with animated keyboard and hands. Best and worst results are highlighted in bold and italic, respectively.

Reference	Entry speed (wpm)	Error rate (%)
Knierim et al. [10]	37.5	7.6
Grubert et al. [11]	34.4	11.5
Hoppe et al. [13]	31.2	*13.4*
Bovet et al. [14]	**44.4**	0.4
Lin et al. [22]	*27.4*	**0.2**

Unfortunately, Table 1 does not provide a clear indication of whether users perform better (in terms of speed) with animated representation of the keyboard and hands. In the studies conducted by Hoppe et al. [13] and Lin et al. [22], users performed much better without any visual feedback compared to when animated keyboard and hands were displayed (34.0 wpm vs. 31.2 wpm [13]; 28.1 wpm vs. 27.4 wpm [22]). However, in the studies conducted by Knierim et al. [10] and Grubert et al. [11], users performed substantially better with visual feedback (31.8 wpm vs. 37.5 wpm [10]; 26.1 wpm vs. 34.4 wpm [11]). We speculate, this is due to the unreliability of the feedback provided in the former studies. It was reported that visual feedback in the former studies was not always accurate or available. Hoppe et al. [13] used a Leap Motion [24] to track hands and reported that it was able to display the hands for about 70% of the time.

Walker et al. [18] and Grubert et al. [11] both provided visual feedback on key-press. However, Walker et al. reported a 40% higher entry speed than Grubert et al. This is likely because Walker et al. used a decoder to enhance the accuracy of their system. They also included an extensive training session before the actual study. Further, in addition to highlighting the currently pressed key like Grubert et al. [11], Walker et al.'s [18] system also highlighted other recently pressed keys to aid in users' sense of proprioception. These could also help explain a 45% lower error rate (15.2% [11] vs. 8.4% [18]). In fact, Walker et al.'s system yielded even a lower error rate with auto-correction (2.6%).

Studies comparing visual feedback through animated fingertips and full hand representation revealed that fingertips yield a slightly higher accuracy rate than full hand [10, 11]. This gain in accuracy is likely due to the obstruction of keys when a full hand is displayed as opposed to fingertips. Interestingly, Grubert et al. [11] identified this difference to be statistically significant ($p < .05$), while Knierim et al. [10] did not.

Table 3. Error rate metrics reported in the literature.

Error metrics	References
Error rate	[2, 10, 11, 13, 14, 21]
Total error rate	[18, 22]

Table 3 presents the error rate metrics reported in the literature. As one can see, most studies reported either the Error Rate (ER) or the Total Error Rate (TER) metrics. However, text entry studies often use different methods to calculate the same error metrics. For instance, Knierim et al. [10] used the Minimal String Distance (MSD) algorithm to count the total number of errors, when Rajanna et al. [25] counted the total number of backspaces. A previous work [9] analyzed these performance metrics and demonstrated how different metrics yield different results. Yet, unfortunately, most studies do not report how they calculate errors.

3 Virtual Qwerty

Numerous virtual text entry techniques have been developed for VR to enable mobility and eliminate the dependence on physical keyboards and keypads. Most of these techniques use the Qwerty layouts or its variants to facilitate the transference of knowledge from the physical to virtual keyboards. These techniques exploit a variety of interaction methods, including head pointing, finger, wrist, and hand gestures, game controllers, touch, eye gazing, and handwriting. The following sections reviews these techniques. Table 4 displays the entry speed and error rates [9] of these techniques from the literature.

3.1 Head Pointing

Since head pointing is the default interaction method for most HMDs, it has been widely used in text entry. Head pointing text entry techniques cast a ray into the scene that is controlled by head movements. A virtual Qwerty keyboard floats in front of users (Fig. 2). To enter a character, users first move the cursor over a key, then select the respective character by either dwelling on it for a predetermined amount of time (a timeout period) [26] or pressing a controller key [26, 27].

Yu et al. [26] evaluated a head pointing technique that used a 400 ms dwell time. It yielded a 10.6 wpm entry speed and 95.8% accuracy rate. Since dwelling can affect the overall entry speed, Majaranta et al. [28] suggested using customizable dwell time. Yu et al. [26] also evaluated a different approach that enables users to select characters by pressing a controller key in place of dwelling. They reported a 15.6 wpm entry speed and 98% accuracy rate by the 6th session. This suggests that head pointing is faster with

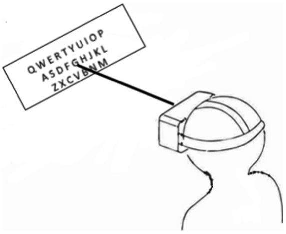

Fig. 2. Text entry through head pointing [27].

keypress than with dwelling. Although these approaches could cause physical discomfort since they force users to constantly move their heads around, could be effective in short-term text entry, such as entering a password or a search keyword [26].

3.2 Finger, Wrist, and Hand Gestures

Many have explored text entry with finger, wrist, and hand gestures in VR. These methods usually use external cameras, sensors, and gloves to track fingers, wrists, and hands.

Bowman et al. [29] developed a digital glove that maps the Qwerty layout to the fingers. Such as, it maps the second-row letters “a”, “s”, “d”, and “f” to the left little, ring, middle, and index fingers, respectively. With this approach, users first rotate their hands to select a row, then enter a character by pinching the thumb and another finger. This method was evaluated in a user study, where it reached on average 6.1 wpm entry speed and 90% accuracy rate [20, 21]. The fact that it lacks haptic feedback may have contributed towards its relatively low speed and accuracy. The KITTY keyboard [30] uses a similar approach, but instead of using hand rotation, it uses the thumb with six degrees of freedom to select a row. Three positions in the front and three in the back of the thumb are assigned to different rows. For example, a pinch between the left little finger and the middle inner thumb enters the letter “a”. This method also lacks haptic feedback. Wu et al. [31] designed a different approach that uses micro speakers to simulate haptic feedback on keypress. With this approach, users wear two data gloves on each hand and enter a character by bending a finger beyond a predetermined threshold.

The main challenge of these techniques is that they demand a substantial amount of time and effort to master. The use of digital gloves makes them a costly solution for both manufacturers and consumers and restricts users from using their hands for secondary tasks. Further, they can strain the fingers when used for an extended period.

Ishii et al. [32] proposed a fist-pointer method, where hand movements and fist gestures are used to select characters. First, users move the pointer by moving the hand in a thumbs-up position. Once the pointer is over the intended character, they select it by folding the thumb. Ishii et al., however, did not evaluate this approach.

Some have also explored mid-air gestures that enables users to select characters by performing hand gestures and finger postures [27, 33]. This technique, too, does not provide haptic feedback. In an evaluation, it yielded a relatively low 9.8 wpm and 92.5% accuracy rate. It was also mentally and physically demanding.

3.3 Controllers

Some techniques enable users to enter text using handheld controllers augmented with motion trackers.

Speicher et al. [27] evaluated four such techniques. The first enables users to use a controller as a laser pointer. Users move the cursor over a keyboard by pointing the controller, then select a character when the cursor is over it by pressing a key. Speicher et al. allowed users to hold two controllers in two hands to facilitate bimanual input. It yielded on average 15.4 wpm and 99% accuracy rate. The second enables users to use a controller as a stylus. Users tap the controller on a character to enter it. It yielded 12.7 wpm and 98.1% accuracy rate. The third enables users to use a controller as a joystick. Users navigate the cursor over a keyboard by pressing the four directional keys, i.e., the four edges of a touchpad, then select a character when the cursor is over it by pressing a key. It yielded 5.3 wpm and 77.2% accuracy rate. The fourth is identical to the third but uses continuous cursor control instead of a discrete movement selection. It yielded 8.4 wpm and 87.8% accuracy rate. While these four techniques are competitive in terms of speed and accuracy, they can cause physical stress in extended use. Results revealed that the latter two techniques were the least physically demanding but the most frustrating due to slower entry speed.

Min et al. [34] designed an ambiguous Qwerty keyboard that arranges the keys into a 3 × 3 grid (Fig. 3). With this approach, users first select a cell by pressing a button, then select the target character by pressing the button once or multiple times, respective to the position of the character in the cell (like Multi-tap [6]). For instance, to enter the letter "p", users first select the top-right cell, then press the button twice. This method saves space due its smaller size, leaving extra space for work, which is rather important in immersive environments [8]. This method has not yet been evaluated.

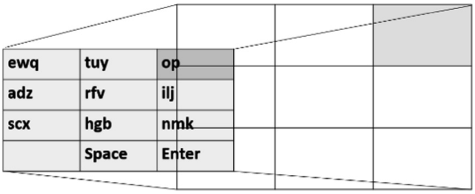

Fig. 3. An ambiguous Qwerty that arranges the keys into a 3 × 3 grid [34].

3.4 Touch-Based Techniques

Some have exploited the popularity of touch-based interaction in VR.

Gugenheimer et al. [35] augmented a 17.78 cm capacitive touchpad on the back of an HMD. They enabled users to enter text by selecting characters on a floating virtual keyboard using the touchpad. They argued that users can use their sense of proprioception to select the correct keys. In an informal evaluation, this approach reached 10 wpm. The challenge with this approach is that it requires users to interact with a trackpad on the back of the HMD, limiting its use to a few scenarios. This method can also cause physical stress when used for an extended period.

Kim and Kim [36] used the hover functionality of a Samsung Galaxy S4 smartphone [37] to enable text entry through its touchscreen. This approach senses the finger using the hover sensor to display its position over a virtual keyboard. Users select a character by either touching the touchscreen or moving the finger beyond the range of the sensor. The former approach enables users to reposition their fingers to correct a selection before leaving the touchscreen. In a user study, these methods reached 7.8 and 9.0 wpm and 79.5% and 92.6% accuracy rate, respectively. The latter approach was more accurate since it enabled users to correct their selections, but at the same time, caused additional physical stress. Although entry speed of these techniques is relatively low, the concept of using hover for text entry in VR is promising.

3.5 Eye Gazing

Many recent HMDs are equipped with eye trackers, making it possible to use eye movements to control the cursor.

Hajana and Ransen [25] studied gaze typing in VR for flat and curved virtual keyboards. They enabled users to select characters by either dwelling on a virtual key for 520 ms or pressing a controller key. The former approach yielded 9.4 and 7.5 wpm, while the latter yielded 10.2 and 9.2 wpm for the flat and curved keyboards, respectively. Accuracy rate for both were over 99%. No significant difference was identified between the curved and flat keyboards.

Ma et al. [38] incorporated a Brain Computer Interface (BCI) with eye gaze for text entry in VR. They combined electric signals from the brain with eye gaze to determine cursor position and selection. Unlike most methods reviewed in this work, this method did not use Qwerty, instead designed an alphabetic layout with 3 rows with 8 characters per row (i.e., first row includes the letters from “a” to “h”). An informal study reported an entry speed of 10.0 wpm, which is relatively low. Yet, this approach could be useful to users with physical disabilities.

3.6 Word-Level Techniques

Word-level text entry techniques have also been explored in VR [26, 39].

Yu et al. [26] investigated gesture typing [40], where users press down a controller key to indicate the start of a gesture, perform the gesture using head movements, then release the button to indicate the end of the gesture. In a user study, this approach reached 24.7 wpm with 94.2% accuracy rate by the 8th session. The accuracy rate of this approach, however, is reliant on the efficiency of its decoder. Further, it has a high physical demand since it requires users to define gesture using expressive head movements.

Popriev et al. [41] and Gonzalez et al. [21] explored handwriting in VR, where users write on an actual tablet using a stylus and the output is displayed on a virtual notepad. This method, however, yielded a low entry speed and accuracy rate, 2.3 wpm and 77%, respectively [21].

4 Novel Virtual Techniques

Many have designed novel keyboard and keypad layouts to facilitate text entry in VR. This section reviews all such techniques.

4.1 Circular and Cubic Layouts

Gonzalez et al. [21] evaluated a circular layout that organizes the letters in an alphabetic order. The keyboard is displayed on a tablet. First, users select a character on the circle using the stylus, then confirm the selection by dragging it to the center of the circle. This approach yielded an entry speed of 4.4 wpm and 98% accuracy rate.

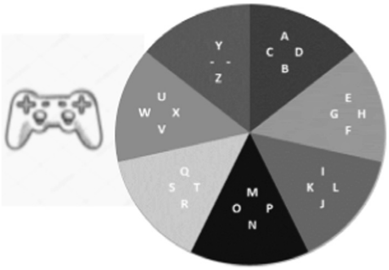

Fig. 4. With the PizzaText keyboard [42], users traverse the keyboard using the right thumbstick and select characters using the left thumbstick.

Yu et al. [42] developed PizzaText, which divides a circle into 7 slices, each containing 4 characters (Fig. 4). Users use the dual thumbsticks of a joystick to interact with this layout [43]—the right thumbstick is used to move around the circular keyboard and the left thumbstick is used to select characters. In a user study, this approach reached 15.9 wpm and 94.6% accuracy rate by the end of the 5th session. Yu et al. [42] evaluated this layout in three different sizes, however, failed to identify a significant difference between the three in terms of speed and accuracy.

4.2 3D Layouts

Most keyboards for VR are in 2D, although virtual environments are in 3D. The Cubic keyboard [44] is a 3D keyboard that arranges the letters in a 3 × 3 × 3 (H × W × D) 3D array. It has 27 cells, 26 for the 26 letters of the English language and a blank cell at the center. Users use a controller to navigate through the cells to select a character. In a pilot study, this approach yielded a competitive entry speed of 21.7 wpm, demanding further exploration of 3D keyboard layouts for VR.

4.3 Hand-Based Techniques

Some have proposed hand-based approaches that map characters onto fingers to reduce the reliance on external hardware.

The BlueTap keyboard [45] maps the letters onto the fingers in an alphabetic order, with at most 4 characters per finger (Fig. 5). Users tap on different parts of the fingers with the thumbs to select characters. This approach uses a wrist worn camera to detect the taps.

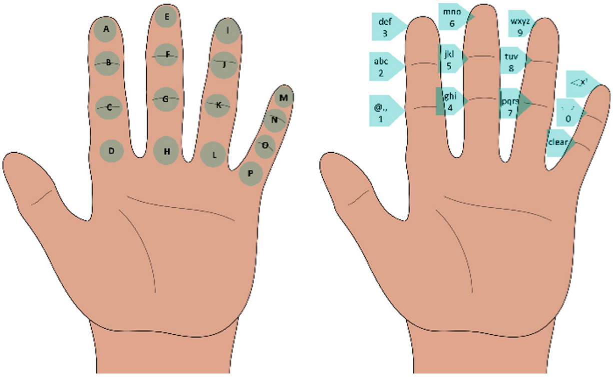

Fig. 5. From left, the BlueTap [45] and the standard 12-key mobile keypad [46] mapped onto the fingers.

Pratorious et al. [46] proposed another approach that maps the standard mobile keypad to the index, middle, and ring fingers. Each knuckle or fingertip includes up to 4 letters (Fig. 5). Like Multi-tap [6], users tap on the knuckle or fingertip once or multiple times, respective to the position of the character. This approach uses a wrist worn camera and an accelerometer to detect the taps. In a pilot study, it yielded 10 wpm.

Ogitani et al. [47] designed a 12-key mobile keypad layout that maps up to 4 characters to each key. With this layout, users tap on the key containing the target character, then swipe towards the direction of the character to enter it (similar to an existing tablet keyboard [48]). They evaluated two techniques using the keypad. The first projects the keypad on the palm and users use the index finger of the other hand to select characters. The second displays the keypad in mid-air and users use their hands to select characters. Both techniques display an animated representation of the hand in the virtual world. These techniques yielded 5.6 wpm and 8.2 wpm, respectively. Interestingly, the authors reported that a projected Qwerty yielded a better entry speed than the keypad, however, did provide further insights into it. We speculate that could be because all participants were familiar with the Qwerty layout.

Although hand-based techniques do not require external devices to function and free up virtual real estate, they do not provide haptic feedback and can cause physical stress. Besides, these techniques force users to perform multiple actions to enter one character, which is a slow approach by design, thus can affect one's overall text entry experience.

Table 4. Virtual text entry techniques with their entry speed in words per minute (wpm) and error rate (%). “Char” signifies “character”. Sensors are not considered as auxiliary devices.

Method	Short description	Input unit	Auxiliary devices	Bimanual	Hand tracking	Participant	Session	Entry speed (wpm)	Error rate (%)
Qwerty [26]	*Users move the cursor over a key via head movement, dwell to enter the respective letter*	Char	✗	✗	✗	6	6	10.6	4.2
	Users move the cursor over a key via head movement, enter the respective char by a keypress		Controller					15.6	3.7
	Users gesture type by moving the cursor via head movement while holding down a key	Word	Controller			12	8	24.7	5.8
Qwerty [20, 21, 29]	*Users enter a char by pinching between a thumb and a finger on the same hand, change rows via hand gestures*	Char	Glove	✓	✓	10	1	6.0 [21]	10.0
Qwerty [30]	*Users enter a char by pinching between a thumb and a finger on the same hand*	Char	Glove	✓	✗	✗	✗	✗	✗
Qwerty [31]	*Provides haptic feedback via a glove on each keypress*	Char	P5 Data Glove	✓	✓	✗	✗	✗	✗
Qwerty [35]	*Users select and enter chars via a touchpad behind the HMD*	Char	Touchpad	✗	✗	3	1	10.0	✗
Qwerty [32]	*Users select a key by moving the hand in a thumbs-up position, enter the respective char by folding the thumb*	Char	✗	✗	✓	✗	✗	✗	✗
Qwertz [27]	*Users enter chars by freehand typing in mid-air*	Char	✗	✓	✓	24	1	9.8	7.6
	Users point at keys using controllers, enter a char by pressing a button* *or performing a push gesture*†		Controller					15.4* 12.7†	1.0* 1.9†
	Users point at keys via direction buttons* *or a touchpad*†*, select a char by pressing a button*							5.3* 8.4†	2.8* 2.2†
	Users point at a key by pointing to it with their head & enter it by pressing a button			✗	✗			10.2	1.2
Qwertz [2]	*Users look straight** *or down*† *to see the keyboard*	Char	Camera	✓	✓	24	1	8.8* 11.0†	3.6* 2.7†
Qwerty [36]	*Users select a key by hovering over the keyboard, enter the respective char by touching** *or leaving*† *the screen*	Char	Phone	✗	✗	10	1	9.0* 7.8†	✗ ✗
Qwerty, Cirrin, Pen [21]	*Users use a pen and tablet to enter chars using Qwerty***, Cirrin*†*, & handwriting*‡	Char	Pen & Tablet	✗	✓	10 28	1	8.2* 2.3† 4.4‡	8.0* 23.0† 2.0‡
Circular PizzaText [42]	*Users use thumbsticks of a controller to enter chars from a circular keyboard containing 7 slices, each with 4 chars*	Char	Controller	✗	✗	10	5	15.9	5.5
BlueTap [45]	*At most four chars are mapped onto all fingers but the thumb, users input by selecting the chars using the thumb*	Char	✗	✓	✓	✗	✗	✗	✗
Qwerty [25]	*Users move the cursor over a flat** *and a curved*† *keyboard via eye movements, enter a char by dwelling for 550* ms	Char	✗	✗	✗	16	4	9.4* 7.5†	0.02* 0.06†
	Users move the cursor over a flat* *and a curved*† *keyboard via eye movements, enter a char by pressing a button*		Controller					10.2* 9.2†	0.07* 0.03†
Mobile Qwerty [34]	*Keys are arranged in a 3 × 3 cell. Users point at a cell via mouse, enter a char using Multi-tap*	Char	Rotational mouse	✓	✓	✗	✗	✗	✗

(*continued*)

Table 4. (*continued*)

Method	Short description	Input unit	Auxiliary devices	Bimanual	Hand tracking	Participant	Session	Entry speed (wpm)	Error rate (%)
A-Z [38]	*Keys are arranged in 3 rows, each containing 8 chars. Eye movements and electric signals from the brains are combined to enter a char*	Char	EEG	✗	✗	✗	✗	10.0	✗
Cubic [44]	*Keys are arranged in a 3 × 3 × 3 (height, width, and depth) grid. Users enter chars using a tracked controller*	Char	Controller	✓	✗	✗	✗	21.6	✗
12-Key Keypad [47]	*Freehand Multi-tap typing in mid-air using a keypad*	Char	✗	✗	✓	6	5	5.2	16.0
	Keypad is projected on one hand, users Multi-tap using the other hand							5.6	22.0
12-Key Keypad [46, 49]	*Keypad is mapped onto the index, middle, ring fingers, users enter chars by tapping on the fingertips & knuckles with the thumb*	Char	✗	✗	✓	✗	✗	10.0	✗

4.4 Discussion

We can see in Table 4 that both Speicher et al. [27] and Yu et al. [26] investigated head pointing coupled with a controller but Yu et al. reported a 35% faster entry speed than Speicher et al. (15.6 wpm vs. 10.2 wpm). We believe the use of a predictive system contributed towards this—Yu et al. augmented their system with predictive features, including word suggestion and completion. They both reported comparable error rates.

Eye gazing [25] and head pointing [26] coupled with dwelling yielded comparable entry speed (9.4 wpm and 10.6 wpm). Eye gazing was slightly faster (11%), most probably due to the use of a shorter dwell time (400 ms vs. 550 ms). The similarly between the two is further established when coupled with a controller. Eye gazing [25] and head pointing [27] both yielded 10.2 wpm when coupled with a controller. This is most probably because both techniques use similar methods for moving the cursor—one uses the head and the other uses eyes. However, it is worth noting that eye gazing has a lower physical demand than head pointing. Yu et al. [25] showed that word-based input using head gestures increases text entry speed by ~40%, it would be worth investigating how this method performs with eye gazing.

Both Speicher et al. [27] and Yu et al. [42] used directional control—the former used thumbsticks with a circular keyboard and the latter used a directional pad with Qwerty. Although the techniques are different, it is worth mentioning that the circular keyboard yielded a 67% higher entry speed than Qwerty (15.9 wpm vs. 5.3 wpm). This is likely due to the compact nature of the circular keyboard layout, which makes it suitable for navigation using directional control. However, the higher error rate of the circular keyboard (5.5% vs. 2.8%) is likely due to the unfamiliarity with the novel layout.

Interestingly, an embossed mobile keypad [21] yielded a 26% faster entry speed than a smartphone virtual Qwerty [36] (12.1 wpm vs. 9 wpm), regardless of the fact that the keypad used Multi-tap. We believe this result could be attributable to the familiarity of feature phones at that time (2009) and the haptic feedback afforded by the keypads.

Table 5. Error rate metrics reported in the literature.

Error metrics	References
Error rate	[2, 21, 25]
Total error rate	[26, 34, 40]
Corrected error rate	[27]

Table 5 presents the error rate metrics reported in the literature. Evidently, most studies reported either the Error Rate (ER) or the Total Error Rate (TER) metrics. The studies that reported TER also reported the Corrected Error Rate (CER) and Uncorrected Error Rate (UER) metrics. Only one study reported only CER [25]. However, text entry studies often use different methods to calculate the same error metrics. Knierim et al. [10], for instance, used the Minimal String Distance (MSD) algorithm to count the total number of errors, when Rajanna et al. [25] counted the total number of backspaces. A prior work [9] analyzed these metrics and showed how different metrics yield different results. Nevertheless, most studies do not report how they calculate errors.

5 Hand Representation

Grubert et al. [11] investigated the effects of different hand representation on text entry in four conditions: none, animated hands, fingertips, and video inlay of the hands. They failed to identify a significant effect of hand representation on entry speed. However, fingertips and video inlay were significantly more accurate. The task load scores for video inlay were also significantly lower. McGill et al. [12] reported similar results. Knierim et al. [10] also compared different hand representations in four conditions: none, realistic hands, abstract hands, and fingertips. Surprisingly, they found out that hand representation did not affect entry speed or accuracy for experienced typists; but affected entry speed for the inexperienced ones. Evidently, inexperienced typists were significantly slower with no hand representation compared to abstract hand representation.

A different study [50] compared male, female, and robotic hand representations. Results revealed that female participants preferred female hands than male and robotic hands, while male participants were mostly neutral.

6 Conclusion

This paper categorized the existing text entry techniques for VR based on their input mechanism. It discussed their strengths, limitations, and the overall performance. It also highlighted important design considerations for the development of more effective text entry techniques. The goal of this paper is to help researchers to comprehend the mechanism of these techniques, compare their performance, and finally identify and address the gaps in this body of work.

6.1 Future Work

This work highlighted the fact that most existing text entry techniques for VR are adaptations of techniques that were designed and optimized for different form factors. As a result, these techniques fail to address all needs and challenges of the paradigm [27]. In VR, speed and accuracy alone do not entirely reflect the effectiveness of a text entry method—usability, learnability, fatigue, and space requirement must also be taken into consideration. Further, as we continue to seek solutions for virtual office spaces, it is important to consider methods that are not only efficient but also portable.

Due to the unviability of an effective text entry technique, physical Qwerty remains an important tool in VR [8]. Although it is substantially faster than most alternatives, it compromises mobility. Further, it relies on expensive sensors to track hands and fingers for visual feedback, which hinders its widespread use. The reliability of the sensors also poses a challenge [13, 27] as most popular tracking devices are error prone [47]. Hence, there is a need for developing cheaper and more reliable tracking devices for physical Qwerty to be fully embraced in VR. Embedding live video streams of the keyboard and hand in VR is an effective alternative [11, 12]. However, further investigation is needed to identify the optimal level of video stream that does not compromise the immersion. The methods for seamlessly blending the virtual and real worlds must also be explored.

In addition, the effects of various keyboard properties have not yet been fully studied. Yu et al. [42] investigated the effects of different sized circular keyboards and Rajana and Hansen [25] investigated different shaped keyboard in gaze typing. However, the effects of the size, position, type (3D vs. 2D), and shape (flat vs. curved) of different types of keyboards in virtual space is still unexplored.

Good user experience is an important ingredient in successful technologies. This review showed that different approaches can be effective in different scenarios [21, 27]. Hence, the possibility of using different text entry solutions for different scenarios must be explored. For instance, users could use a pointing-based or an ambiguous technique for short-term text entry, then switch to a physical keyboard for heavy text entry sessions. It is also essential that we design alternative text entry techniques specifically for 3D environments, such as the Cubic keyboard [44]. Circular and touch-based keyboards also demand further exploration [36, 42].

References

1. 2019-2024 Global and regional augmented reality and virtual reality industry production sales and consumption status and prospects professional market research report. https://www.360marketupdates.com/2019-2024-global-and-regional-augmented-reality-and-virtual-reality-industry-production-sales-and-consumption-status-and-prospects-professional-market-research-report-13733991
2. Grubert, J., Witzani, L., Ofek, E., Pahud, M., Kranz, M., Kristensson, P.O.: Text entry in immersive head-mounted display-based virtual reality using standard keyboards, pp. 1–8 (2018)
3. Grubert, J., Ofek, E., Pahud, M., Kristensson, P.O.: The office of the future: virtual, portable, and global. IEEE Comput. Graph. Appl. **38**, 125–133 (2018). https://doi.org/10.1109/MCG.2018.2875609

4. Nguyen, C., DiVerdi, S., Hertzmann, A., Liu, F.: CollaVR. In: Proceedings of the 30th Annual ACM Symposium User Interface Software Technology - UIST 2017, pp. 267–277 (2017). https://doi.org/10.1145/3126594.3126659
5. Ma, C., Du, Y., Teng, D., Chen, J., Wang, H., Dai, G.: An adaptive sketching user interface for education system in virtual reality. ITME2009 – Proceedings of the 2009 IEEE International Symposium on IT Medical Education, pp. 796–802 (2009). https://doi.org/10.1109/ITIME.2009.5236314
6. Arif, A.S.: Predicting and reducing the impact of errors in character-based text entry (2013). http://hdl.handle.net/10315/28170
7. Salthouse, T.A.: Effects of age and skill in typing. J. Exp. Psychol. Gen. **113**, 345–371 (1984). https://doi.org/10.1037/0096-3445.113.3.345
8. LaViola Jr., J.J., Kruijff, E., McMahan, R.P., Bowman, D., Poupyrev, I.P.: 3D User Interfaces: Theory and Practice. Addison-Wesley Professional, Boston (2017)
9. Arif, A.S., Stuerzlinger, W.: Analysis of text entry performance metrics. In: Proceedings of the IEEE Toronto International Conference - Science and Technology for Humanity - TIC-STH 2009, pp. 100–105. IEEE (2009). https://doi.org/10.1109/TIC-STH.2009.5444533
10. Knierim, P., Schwind, V., Feit, A.M., Nieuwenhuizen, F., Henze, N.: Physical keyboards in virtual reality: analysis of typing performance and effects of avatar hands. In: Proceedings of the 2018 CHI Conference on Human Factors Computing Systems, pp. 1–9 (2018). https://doi.org/10.1145/3173574.3173919
11. Grubert, J., Witzani, L., Ofek, E., Pahud, M., Kranz, M., Kristensson, P.O.: Effects of hand representations for typing in virtual reality, pp. 1–8 (2018)
12. McGill, M., Boland, D., Murray-Smith, R., Brewster, S.: A Dose of Reality: Overcoming Usability Challenges in VR Head-Mounted Displays. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 2143–2152 (2015). https://doi.org/10.1145/2702123.2702382
13. Hoppe, A.H., Otto, L., van de Camp, F., Stiefelhagen, R., Unmüßig, G.: qVRty: virtual keyboard with a haptic, real-world representation. In: Stephanidis, C. (ed.) HCI 2018. CCIS, vol. 851, pp. 266–272. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-92279-9_36
14. Bovet, S., Curran, N., Kehoe, A., Gutierrez, M., Rouvinez, T., Crowley, K.: Using traditional keyboards in VR: SteamVR developer kit and pilot game user study. In: The 2018 IEEE Games, Entertainment, Media Conference - GEM 2018, pp. 131–134. IEEE (2018)
15. Introducing the Logitech BRIDGE SDK. https://blog.vive.com/us/2017/11/02/introducing-the-logitech-bridge-sdk/
16. VIVE Pro - The Professional-grade VR Headset. https://www.vive.com/us/product/vive-pro
17. Kim, S., Kim, G.J.: Using keyboards with head mounted displays. In: Proceedings of the VRCAI 2004 - ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry, pp. 336–343 (2004). https://doi.org/10.1145/1044588.1044662
18. Walker, J., Li, B., Vertanen, K., Kuhl, S.: Efficient typing on a visually occluded physical keyboard. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI 2017. pp. 5457–5461. ACM Press, New York (2017). https://doi.org/10.1145/3025453.3025783
19. Vertanen, K., Memmi, H., Emge, J., Reyal, S., Kristensson, P.O.: VelociTap: investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems - CHI 2015, pp. 659–668 (2015). https://doi.org/10.1145/2702123.2702135

20. Bowman, D.A., Rhoton, C.J., Pinho, M.S.: Text input techniques for immersive virtual environments: an empirical comparison. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 46, pp. 2154–2158 (2002). https://doi.org/10.1177/154193120204602611
21. González, G., Molina, J.P., García, A.S., Martínez, D., González, P.: Evaluation of text input techniques in immersive virtual environments. In: Macías, J., Granollers Saltiveri, A., Latorre, P. (eds.) New Trends on Human-Computer Interaction, pp. 109–118. Springer, (2009). https://doi.org/10.1007/978-1-84882-352-5_11
22. Lin, J.-W., et al.: Visualizing the keyboard in virtual reality for enhancing immersive experience. In: ACM SIGGRAPH 2017 Posters - SIGGRAPH 2017, vol. 1, pp. 1–2 (2017). https://doi.org/10.1145/3102163.3102175
23. P.I. Engineering: X-keys Computer Input Devices. https://xkeys.com/xkeys.html
24. Leap Motion. https://www.leapmotion.com/
25. Rajanna, V., Hansen, J.P.: Gaze typing in virtual reality: impact of keyboard design, selection method, and motion. In: Proceedings of the 2018 ACM Symposium on Eye Tracking Research and Applications - ETRA 2018, pp. 1–10. ACM Press, New York (2018). https://doi.org/10.1145/3204493.3204541
26. Yu, C., Gu, Y., Yang, Z., Yi, X., Luo, H., Shi, Y.: 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 2017, pp. 4479–4488. ACM Press, New York (2017). https://doi.org/10.1145/3025453.3025964
27. Speicher, M., Feit, A.M., Ziegler, P., Krüger, A.: Selection-based text entry in virtual reality. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI 2018, pp. 1–13. ACM Press, New York (2018). https://doi.org/10.1145/3173574.3174221
28. Majaranta, P., Ahola, U., Špakov, O.: Fast gaze typing with an adjustable dwell time. In: Proceedings of the 27th international conference on Human factors in computing systems - CHI 2009, p. 357. ACM Press, New York (2009). https://doi.org/10.1145/1518701.1518758
29. Bowman, D.A , Ly, V.Q., Campbell, J.M.: Pinch keyboard: natural text input for immersive virtual environments. Virginia Tech Department of Computer Science Technical report (2001)
30. Mehring, C., Kuester, F., Singh, K.D., Chen, M.: KITTY: keyboard independent touch typing in VR. In: Proceedings - Virtual Reality Annual International Symposium, pp. 243–244 (2004). https://doi.org/10.1109/VR.2004.1310090
31. Wu, C.-M., Hsu, C.-W., Lee, T.-K., Smith, S.: A virtual reality keyboard with realistic haptic feedback in a fully immersive virtual environment. Virtual Real. **21**, 19–29 (2017). https://doi.org/10.1007/s10055-016-0296-6
32. Ishii, A., Adachi, T., Shima, K., Nakamae, S., Shizuki, B., Takahashi, S.: FistPointer: target selection technique using mid-air interaction for mobile VR environment. In: Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems, p. 474 (2017). https://doi.org/10.1145/3027063.3049795
33. Markussen, A., Jakobsen, M.R., Hornbæk, K.: Selection-based mid-air text entry on large displays. In: Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., Winckler, M. (eds.) INTERACT 2013. LNCS, vol. 8117, pp. 401–418. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40483-2_28
34. Min, K.: Text input tool for immersive vr based on 3 × 3 screen cells. In: Lee, G., Howard, D., Ślęzak, D. (eds.) ICHIT 2011. LNCS, vol. 6935, pp. 778–786. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24082-9_94

35. Gugenheimer, J., Dobbelstein, D., Winkler, C., Haas, G., Rukzio, E.: FaceTouch: enabling touch interaction in display fixed uis for mobile virtual reality. In: Proceedings of the ACM User Interface Software and Technology (UIST 2016), 49–60 (2016). https://doi.org/10.1145/2984511.2984576
36. Kim, Y.R., Kim, G.J.: HoVR-type: smartphone as a typing interface in VR using hovering. In: 2017 IEEE International Conference on Consumer Electronics, ICCE 2017, pp. 200–203 (2017). https://doi.org/10.1109/ICCE.2017.7889285
37. Petrovan, B.: Galaxy S4 to feature floating touch, highly efficient AMOLED display. https://www.androidauthority.com/galaxy-s4-floating-touch-display-167326/
38. Ma, X., Yao, Z., Wang, Y., Pei, W., Chen, H.: Combining brain-computer interface and eye tracking for high-speed text entry in virtual reality. In: Proceedings of the 2018 Conference on Human Information Interaction - IUI 2018. Part F1351, pp. 263–267 (2018). https://doi.org/10.1145/3172944.3172988
39. Jimenez, J.G., Schulze, J.P.: Continuous-motion text input in virtual reality. Electron. Imaging 450-1–450–6 (2018). https://doi.org/10.2352/ISSN.2470-1173.2018.03.ERVR-450
40. Alvina, J., Malloch, J., Mackay, W.E.: Expressive keyboards: enriching gesture-typing on mobile devices. In: Proceedings of the 29th Annual Symposium on User Interface Software and Technology - UIST 2016, pp. 583–593. ACM Press, New York (2016). https://doi.org/10.1145/2984511.2984560
41. Poupyrev, I., Tomokazu, N., Weghorst, S.: Virtual notepad: handwriting in immersive VR. In: Proceedings of the IEEE 1998 Virtual Reality Annual International Symposium (Cat. No.98CB36180), pp. 126–132 (1998). https://doi.org/10.1109/VRAIS.1998.658467
42. Yu, D., Fan, K., Zhang, H., Monteiro, D., Xu, W., Liang, H.: PizzaText: text entry for virtual reality systems using dual thumbsticks. IEEE Trans. Vis. Comput. Graph. **24**, 2927–2935 (2018). https://doi.org/10.1109/TVCG.2018.2868581
43. Wilson, A.D., Agrawala, M.: Text entry using a dual joystick game controller. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 475–478. ACM, New York (2006). https://doi.org/10.1145/1124772.1124844
44. Yanagihara, N., Shizuki, B.: Cubic keyboard for virtual reality. In: Proceedings of the Symposium on Spatial User Interaction - SUI 2018, pp. 170–170. ACM Press, New York (2018). https://doi.org/10.1145/3267782.3274687
45. BlueTap—The Ultimate Virtual-Reality (VR) Keyboard. https://medium.com/eunoia-i-o/bluetap-the-ultimate-virtual-reality-vr-keyboard-77f1e3d57d6f
46. Prätorius, M., Valkov, D., Burgbacher, U., Hinrichs, K.: DigiTap: an eyes-free VR/AR symbolic input device. In: Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology, pp. 9–18. ACM, New York (2014). https://doi.org/10.1145/2671015.2671029
47. Ogitani, T., Arahori, Y., Shinyama, Y., Gondow, K.: Space saving text input method for head mounted display with virtual 12-key keyboard. In: 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications, pp. 342–349 (2018). https://doi.org/10.1109/AINA.2018.00059
48. Arif, A.S., Pahud, M., Hinckley, K., Buxton, B.: Experimental study of stroke shortcuts for a touchscreen keyboard with gesture-redundant keys removed. In: Proceedings of the 2014 Graphics Interface Conference - GI 2014, pp. 43–50. Canadian Information Processing Society (2014). https://doi.org/10.20380/GI2014.06

49. Prätorius, M., Burgbacher, U., Valkov, D., Hinrichs, K.: Sensing thumb-to-finger taps for symbolic input in VR/AR environments. IEEE Comput. Graph. Appl. **35**, 42–54 (2015). https://doi.org/10.1109/MCG.2015.106

50. Schwind, V., Knierim, P., Tasci, C., Franczak, P., Haas, N., Henze, N.: These are not my hands! In: Proceedings of 2017 CHI Conference on Human Factors in Computing Systems - CHI 2017, pp. 1577–1582 (2017). https://doi.org/10.1145/3025453.3025602