International Journal of Human–Computer Interaction

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/hihc20

Light-Occlusion Text Entry in Mixed Reality

Aoxin Sun, Lili Wang, JiaYe Leng & Sio Kei Im

INTERNATIONAL JOURNAL OF HUMAN–COMPUTER INTERACTION
https://doi.org/10.1080/10447318.2023.2285646

Light-Occlusion Text Entry in Mixed Reality

Aoxin Sun[a], Lili Wang[a,b], JiaYe Leng[c], and Sio Kei Im[d]

[a]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China; [b]Peng Cheng Laboratory, Shengzhen, China; [c]School of Creative Media, City University of Hong Kong, Hong Kong, China; [d]Faculty of Applied Sciences, Macao Polytechnic University, Macau, China

ABSTRACT

Text entry is a recurring task in mixed reality (MR) applications, and the ability of eyes-free text entry methods to allow users to enter text without focusing on the input device is ideal and compelling. However, existing eyes-free text entry methods leave much to be desired regarding efficiency and accuracy. In this paper, we propose a new light-occlusion text entry method in MR environment that uses dual thumb typing on a touchscreen. We design a partially visible keyboard as visual feedback to improve user performance. In addition, we optimize the underlying keyboard by collecting eyes-free typing data through a user study. The results show that our method has high typing speed, low error rate, and is very novice-friendly. After a short training period, the average typing speed of the novice group can reach 26.23 WPM (words per minute), while the average typing speed of the potential expert group can reach 30.62 WPM.

KEYWORDS

Human-computer interaction; mixed reality; text entry; touchscreen; occlusion

1. Introduction

In recent years, mixed reality (MR) has been widely used in various fields such as healthcare, education, and entertainment. With the in-depth application of mixed reality and the introduction of related concepts such as "metaverse," the demand for text entry in mixed reality is also increasing, such as note-taking, chatting, file name input, and input tasks in games. Therefore, how to input text efficiently and correctly in MR environments has become an important issue. However, MR devices are usually mobile and wearable and can only be operated by wearing HMDs and bare hands. Therefore, it is more challenging to input text efficiently.

Many researchers have explored the efficiency, learnability, and usability of touchscreen-based eyes-free text entry methods. Blindtype (Y. Lu et al., 2017) is a touchscreen single-thumb input method that utilizes muscle memory to achieve input speeds of 17–23 WPM. i's Free (S. Zhu et al., 2019) is a touchscreen single-thumb text entry method that decodes the user's sliding gestures and has an input speed of 22 WPM. TOAST (Shi et al., 2018) uses 10-finger for input and achieves a speed of 44.6 WPM, but is not applicable to mobile application scenarios. These techniques are a good validation of the feasibility of eyes-free input on touchscreens. However, existing touchscreen-based approaches are either inefficient with one-finger input, or 10-finger input which cannot be adapted to mobile application scenarios.

Methods such as iText (X. Lu et al., 2021) are eyes-free methods without any visual feedback about the keyboard, which may confuse users and are not easy to correct. Blindtype and text entry on Hololens have a full keyboard for visual feedback, and it is difficult for users to efficiently access the information of the real scene because of the large occlusion. Another option for text entry in MR is the hands-free method, and they usually use eye or head movements to control the input. However, the main problem with hands-free text entry is that it is difficult for the user to master, and the typing speed is generally low. For example, iText (X. Lu et al., 2021) uses the user's eye movements to enter text, and its maximum typing speed is only 13.76 WPM. RingText (Xu et al., 2019) uses the user's head movements to control the virtual cursor for text input, and its maximum typing speed is only 13.24 WPM.

Inspired by the above work, we think of an idea of light-occlusion text entry, i.e., first exploring using two thumbs on a mobile phone touchscreen for eyes-free text entry and then employing a light-occlusion keyboard visual feedback design to make text entry easier. For example, when a user replies to a message or transcribes text content while walking, they will not feel excessive visual occlusion from the keyboard. We choose the mobile phone as the interaction device that almost everyone has. Users can take out their phones to enter text when needed, put their phones in their pockets when not needed, free up their hands for other things, and increase their typing speed compared to one-handed typing.

In this paper, we propose an efficient light-occlusion text entry method that allows users to enter text on a mobile phone touchscreen using two thumbs. We designed a partially visible keyboard that provides visual feedback and

CONTACT Lili Wang wanglily@buaa.edu.cn; sunaoxin@buaa.edu.cn State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China; Peng Cheng Laboratory, Shengzhen, China

allows users to enter text with less occlusion. We collected user typing data and used an intra-block optimization approach to adjust the layout of the mobile phone's underlying keyboard. Finally, we designed a 5-day user study to evaluate the performance of our approach. The results show that our method is efficient, accurate, and novice-friendly. Novice users can achieve an average of 26.23 WPM (s.e. = 0.69), and expert users can achieve an average of 30.62 WPM (s.e. = 1.61) on the fifth day. The average NCER and TER for the five days were 0.89% (s.e. = 0.29%) and 3.24% (s.e. = 0.65%) respectively. After five days of training, novices improved their typing speed by 47%, while potential experts improved their typing speed by 17%. Compared to Blindtype and i's Free, the typing speed of our method increased by about 25% (Figure 1).

In summary, the contributions of our light-occlusion text entry are as follows:

- to the best of our knowledge, we are the first ones to explore two-thumb eyes-free input on the mobile phone touchscreen;
- we designed a partially visible keyboard to provide visual feedback for text entry and to reduce occlusion in mixed-reality environments;
- we propose an intra-block optimization method to adjust the layout of the underlying QWERTY keyboard on the touchscreen;
- we designed a user study to evaluate the performance of our text entry method.

2. Related work

In this section, we review the existing text entry methods in MR and touchscreen-based text entry methods.

2.1. Text entry in MR

MR includes Virtual Reality and Augmented Reality (VR and AR). Text entry methods in VR environments have been widely explored. The most widely used interaction device in existing VR systems is the controller. Many researchers have explored the performance of controller-based text entry methods. Flower Text Entry (Leng et al., 2022) is a controller-based text entry method that uses a flower-shaped keyboard and incorporates "3D hand interaction." In the HiPad (Jiang & Weng, 2020) method, the user uses a controller with a circular touch screen for virtual keyboard text entry. The layout of the HiPad's virtual keyboard is based on circles and squares with rounded corners, with the outer part of the circle being the area of the six keys containing the letters of the alphabet. PizzaText (Yu et al., 2018) is a method of text entry using a circular keyboard layout, which divides a circle into six blocks, each containing four characters. Drum-like keyboards (Boletsis & Kongsvik, 2019) treat the controller rays as "drumsticks" that are moved downward to "press" the keys of the virtual keyboard. While controller-based text entry methods are widely used, these methods still require a full keyboard display to provide visual feedback and are not novice-friendly. The KSPC (keystrokes per character) metric (MacKenzie, 2002) is defined as the average number of keystrokes required to type each character in a particular language using a particular input method. Optimally, the KSPC should be 1, while controller-based methods have a KSPC mostly greater than 1, which can lead to a decrease in input speed.

Compared to VR applications, AR applications require more mobility and hands-free operation from the user. There are some text entry methods in AR environments departing from the traditional concept of "typing by hand," which take advantage of other parts of the human body for typing. SWIFTER (Pick et al., 2016) is a speech-based multimodal typing technique, and experimental results show that it has good user acceptance. Adhikary and Vertanen (2021) incorporate speech recognition techniques into mid-air typing. Their approach takes sentences as input and corrects errors by selecting speech alternatives provided by a speech recognizer. iText (X. Lu et al., 2021) is a text entry method in AR environment by capturing the user's eye positions and eye movements. Gaze Speedup (Zhao et al., 2023) is a mid-air gesture typing method that speeds up the gesture cursor toward the user's gaze fixation location. Lystbæk et al. (2022) investigated gaze-assisted selection-based text entry through the concept of spatial alignment of both modalities. All these methods have great potential, but there are still some problems: the eye-based methods need to be improved in terms of typing speed and may cause visual fatigue as well as motion sickness with prolonged use; the speech-based methods achieve good performance but can be problematic in certain situations, such as noise and privacy

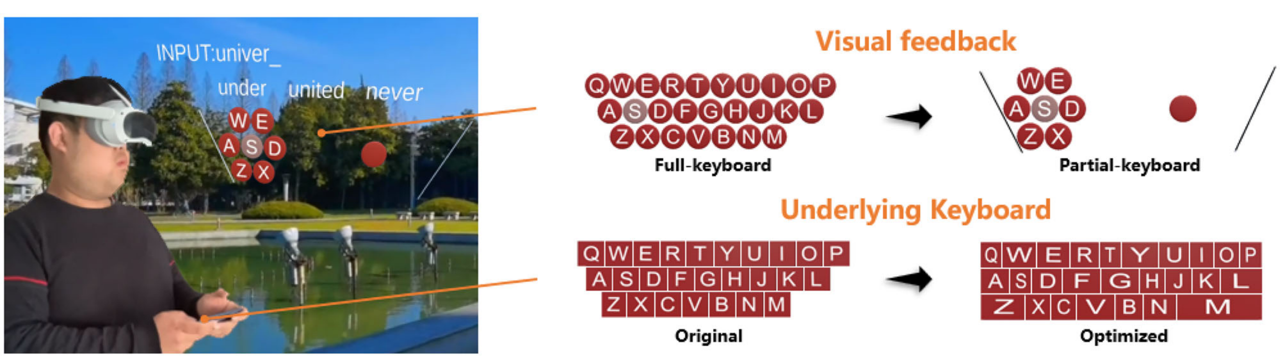


Figure 1. In mixed reality, the user holds a mobile phone and applies our text entry method for typing.

in shared environments (Shneiderman, 2000) and error correction problems (Vertanen, 2009). We propose an efficient and accurate method that can be applied to both VR and AR environments.

2.2. Text entry with touchscreen

Touchscreen-based approaches have shown good performance in mobile scenarios where the user inputs via an external touchscreen. The current touchscreen technique focuses on "eyes-free" methods. Typically, the most familiar method of eyes-free text entry for users is to use a physical keyboard for text entry, which can reach an input speed of 60–100 WPM. However, MacKenzie et al. (1999) found that the speed of eyes-free text entry on a touchscreen can only reach 20 WPM. Wang et al. (2013) found that the accuracy rate of the eyes-free text entry method is only 85%. Therefore, it is necessary to include some optimization algorithms. Sun et al. (2019) explored the low occlusion QWERTY soft keyboard using spatial landmarks, which only give some dots and segments to indicate the position of special characters in the QWERTY soft keyboard rather than any part of the keyboard. After five 15-phrase typing sessions, participants achieved 88.1%–92.8% of the full QWERTY soft keyboard in terms of WPM.

Blindtype (Y. Lu et al., 2017) has two statistical-based word-level decoding algorithms: "absolute algorithm" and "relative algorithm." Both algorithms achieve superior performance with typing speeds of 17–23 WPM. i's Free (S. Zhu et al., 2019) is an algorithm for decoding user swipe gestures. The technique vectorizes the user's swipe gestures and decodes them using a dynamic time-warping algorithm (Sakoe & Chiba, 1978). The expert users can achieve a typing speed of 22 WPM after 10 min of training. GlanceType (Y. Lu et al., 2019) is a text entry method using a spilt soft keyboard on a tablet computer. The evaluation shows that the method substantially improves the performance of text entry. Bi et al. (2012) explored a bimanual gesture soft keyboard on a tablet computer. The results indicated that the new gesture keyboards were valuable complements to unimanual gestures. Azenkot et al. (2012) proposed Perkinput, a 6-bit Braille text entry on the touchscreen that could reach 17.56 WPM for one hand and 38.0 WPM for two hands after training. Southern et al. (2012) explored the efficiency of the Perkinput on mobile phones, and after 20 min of training, blind expert users reached 23.2 WPM.

Touchscreen-based typing is undoubtedly easy to learn and highly efficient, but there are still problems with existing touchscreen-based methods. Some methods require a full keyboard for visual feedback, and the keyboard interface in the MR environment will affect the user's experience. Also, the absence of visual feedback from the keyboard may confuse the user while typing. To balance the impact of the above two types of keyboards on the visual experience, we propose a method to display a partial keyboard for visual feedback. We also notice that the current touchscreen-based method is either a single-finger method on a small touchscreen or a multi-finger method on a large touchscreen, while we focus on using two thumbs to type on a mobile phone, which considers both efficiency and portability.

3. Design rationale

We aim to design a light-occlusion method that balances speed, accuracy, and learnability while mitigating occlusion problems caused by the keyboard interface in MR environments. We will introduce the design rationales from the following aspects.

3.1. Light occlusion

Many eyes-free text entry methods have been augmented with visual feedback to improve input efficiency and accuracy. Blindtype (Y. Lu et al., 2017) incorporates a full QWERTY keyboard to help users perform text entry. i's Free (S. Zhu et al., 2019) also displays a full QWERTY keyboard to help users recall after their fingers have been off the screen for a period of time (300 ms). These methods require a full keyboard interface, thus preventing the user from observing the scene in front of them during typing, and these methods can be considered to provide fully occluded keyboard interfaces. There are also methods (X. Lu et al., 2021) that attempt to allow the user to type on an imaginary keyboard, using a none-keyboard interface to solve the problem. However, such methods would affect users' self-confidence in typing because they do not have access to specific keystroke information during the typing process.

Therefore, the main point when we design the keyboard interface is a partially occluded interface. We consider designing a partial keyboard visual feedback method, which can avoid visual occlusion in MR environments, and at the same time can achieve a similar or even better feedback effect than the full QWERTY keyboard. Including spatial landmarks can also be considered to improve the user's confidence when performing light-occlusion text entry.

3.2. Efficiency and accuracy

Related studies have shown that the accuracy rate of one-handed eyes-free text entry is only 85% (Wang et al., 2013). Therefore, we focus on two-handed typing, which can improve the efficiency and accuracy rate to some extent. In addition, reasonable visual guidance (e.g., displaying a partial keyboard) can help the user confirm whether the current selection is what he/she expects, and can improve efficiency and accuracy by swiping on the screen to make corrections when the user finds errors. Word correction can help users automatically correct spelling errors, significantly reducing the input error rate. When designing the text entry technique, we should consider integrating a word correction function. Auditory feedback during text entry can also help users confirm information and thus improve the accuracy rate. Finally, the left and right key areas should be as balanced as possible when typing with both hands to ensure that the number of letters is similar or equal. This can avoid excessive fatigue in one hand when typing for long periods.

3.3. Familiarity

Many methods (Boletsis & Kongsvik, 2019; Leng et al., 2022; Y. Lu et al., 2017; Shi et al., 2018; S. Zhu et al., 2019) have demonstrated that the most familiar keyboard layout for users is the QWERTY keyboard, and users can recall the general layout of the QWERTY keyboard when they are typing. Therefore, we can consider the following three rules when designing: firstly, the keys of the partial keyboard should keep their relative positions, which allows users to make corrections in time for error clicks; secondly, the display of the partial keyboard should be able to inform the user of the current position of the touch point in the overall keyboard, to facilitate the user's position guidance during continuous input; thirdly, when using a QWERTY keyboard to input, the keyboard is usually divided into two parts: left-hand area and right-hand area. We also take this into account when optimizing the underlying keyboard. We divided the underlying keyboard into six blocks, and we adopted an intra-block modification strategy when applying our keyboard optimization algorithm. In addition, our method uses two thumbs for text entry, which is familiar to users because they often use two thumbs to play games or watch videos on their mobile phones.

4. Text entry leveraging touchscreen

In this section, we introduce the design ideas of the choice of input devices, methods, and underlying keyboard design.

4.1. Input devices and methods

As already mentioned, we use a mobile phone as the input device and choose two-thumb tapping, facilitating text entry independent of time and place. The small size of mobile phones means that 10-finger touch typing is impossible. Instead, the input uses a single index finger, a single thumb, or two thumbs (Ye et al., 2020). Azenkot and Zhai (2012) reported that text entry speed reached approximately 36 WPM with a single index finger, 33 WPM with one thumb, and 50 WPM with two thumbs. Two-thumb input is the most efficient of these three input methods. In addition, using a two-thumb keyboard makes good use of screen space. Another benefit of two-thumb operation is mobility, eliminating the need to move fingers from one side to the other frequently. Tapping is still the preferred mode of touchscreen keyboard operation for many users today. Gesture keyboards should not be viewed as competition or as a replacement for tapping. They should be compatible with the tapping mode of the keyboard as well (Bi et al., 2012). Moreover, the current mainstream swiping keyboards are based on one hand, and two-handed gestures are challenging to implement.

4.2. Underlying keyboard design

The current mobile phone screen size is 6–7 inches, and the screen aspect ratio is about 2:1. Our underlying keyboard can automatically be adapted to the specific size and aspect ratio of the mobile phone. We used a Redmi Note11 running Android 11 as the input device. The device screen size is 6.6 inches and measures 163.56×75.78 mm. MR HMDs and mobile phones communicate using sockets. For use on large touchscreen devices such as the iPad, the split keyboard can be adopted (Bi et al., 2012; Y. Lu et al., 2019). Our underlying keyboard is shown in Figure 2, which can be roughly divided into four areas: the standard QWERTY keyboard area, the space area, the backspace area, and the word prediction/correction selection area. The underlying keyboard is not displayed on the phone. We have also made initial adjustments to the underlying keyboard. The keys of "A," "Z," "L," and "M" are stretched to the screen edges to take advantage of the infinite size of the edges according to Fitz's law (MacKenzie, 2018). Word-level input correction is one of the most widely used mechanisms in modern smart keyboards. We use SymSpell (Garbe, 2019) to predict the subsequent letters of a word based on the user's input for word completion, which can also correct spelling errors. It improves typing speed and reduces error rates (Leng et al., 2022). When the user types, the first three predicted results are displayed in the candidate area. When the user selects a word, the corresponding word in the candidate area will be red.

5. Partially visible keyboard

In this section, we design a partial keyboard for visual feedback. As a partially occluded keyboard interface, we compare it with non-occluded and fully occluded types of keyboard interfaces to explore their impact on typing performance. We designed a pilot user study for the evaluation.

5.1. Partial keyboard design

We have drawn on the idea of a circular keyboard layout of related methods (Jiang & Weng, 2020; Leng et al., 2022; Xu et al., 2019; Yu et al., 2018) and designed a partially visible keyboard shown in Figure 3a. We display the currently touched key in the center of the partial keyboard, and in the surrounding six corners, we display the keys adjacent to the touched key and no other keys. The number of displayed keys is automatically adjusted when the touch point is on the boundary of the full keyboard. Each key has a diameter of 0.2 meters, and the keyboard is fixed 1.8 meters in front of the camera.

There is considerable overlap between keys when users perform eyes-free text entry, but this overlap is not more than one key at most, i.e., errors always occur between adjacent keys. Given the above problem, as shown in Figure 3a, our method provides the user with both the current key and the surrounding keys, which can help the user quickly correct errors. If the current key is what the user expects, it can

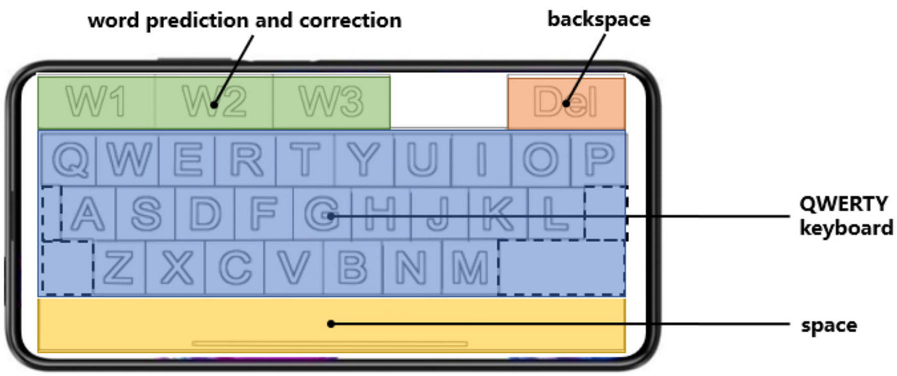


Figure 2. The underlying keyboard model on mobile phone.

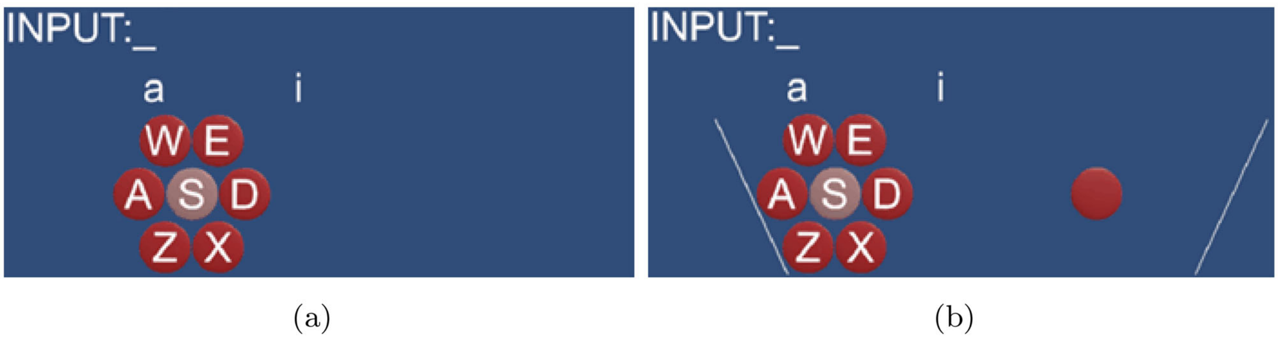


Figure 3. Partial keyboard is visualized in (a) and spatial landmarks are visualized in (b).

be confirmed directly, and if there is an error, then the desired key can be selected by sliding the finger. When the user's finger leaves the screen, the key touched when the finger left will be entered.

5.2. Spatial landmarks

Inspired by previous work (Sun et al., 2019), we also add simple lines and dots (as shown in Figure 3b) as spatial landmarks on the partial keyboard. Our initial keyboard interface had no boundary definitions, which could have led to confusion. We add symmetrical white lines on the left and right sides, which are the boundaries of the keyboard, and two dots centered on the left area of the keyboard (the position of "D") and the right area of the keyboard (the position of "J"), respectively.

5.3. Pilot user study 1: Evaluate the impact of visual feedback on typing performance

In subsection 5.1 and subsection 5.2, we proposed a partial keyboard along with spatial landmarks, whose primary purpose is to balance accuracy and reduce visual occlusion. In this section, we design a user study comparing it with other common forms of virtual keyboards to evaluate the impact of visual feedback on typing performance. We compare our partial keyboard with three typical virtual keyboards: none-keyboard, full-keyboard, and transparent full-keyboard, demonstrating that our partial keyboard is more efficient while effectively reducing occlusion.

5.3.1. User study design

5.3.1.1. Participants. Sixteen participants (10 males and six females, aged 21–25) from our university were recruited for this study. They are all familiar with the basic layout of the QWERTY keyboard. None of them have experienced text entry in MR.

5.3.1.2. Hardware setup. A Redmi Note11 running Andriod 11 is used as the touchscreen to run the input program. The screen size is 6.6-inch and measures 163.56×75.78 mm. A PICO4 is used to display the visual feedback and the input box. MR HMDs and mobile phones communicate using sockets. When the user touches a key on the phone, the MR HMDs give the appropriate feedback. The programs are developed in C# in Unity 2021.3.

5.3.1.3. Task and procedure. This study consisted of five sessions, and all participants were required to attend five sessions. The task was to enter 10 phrases in each session. The order of the sessions was counterbalanced across participants, and phrases in each session were randomly generated from the Mackenzie phrase set (MacKenzie & Soukoreff, 2003). Participants were asked to enter the text "quickly and accurately." As shown in Figure 4, The experimental environment was a virtual classroom where the target phrase was written on a virtual blackboard directly in front of the user. The user's initial line of sight, the keyboard display area, and the target phrase were in a straight line. Users can reduce the occlusion by moving their heads. Session 1 is the baseline with none-occlusion (Figure 4a). Session 2 is with a full QWERTY keyboard (Figure 4b). Session 3 is with a

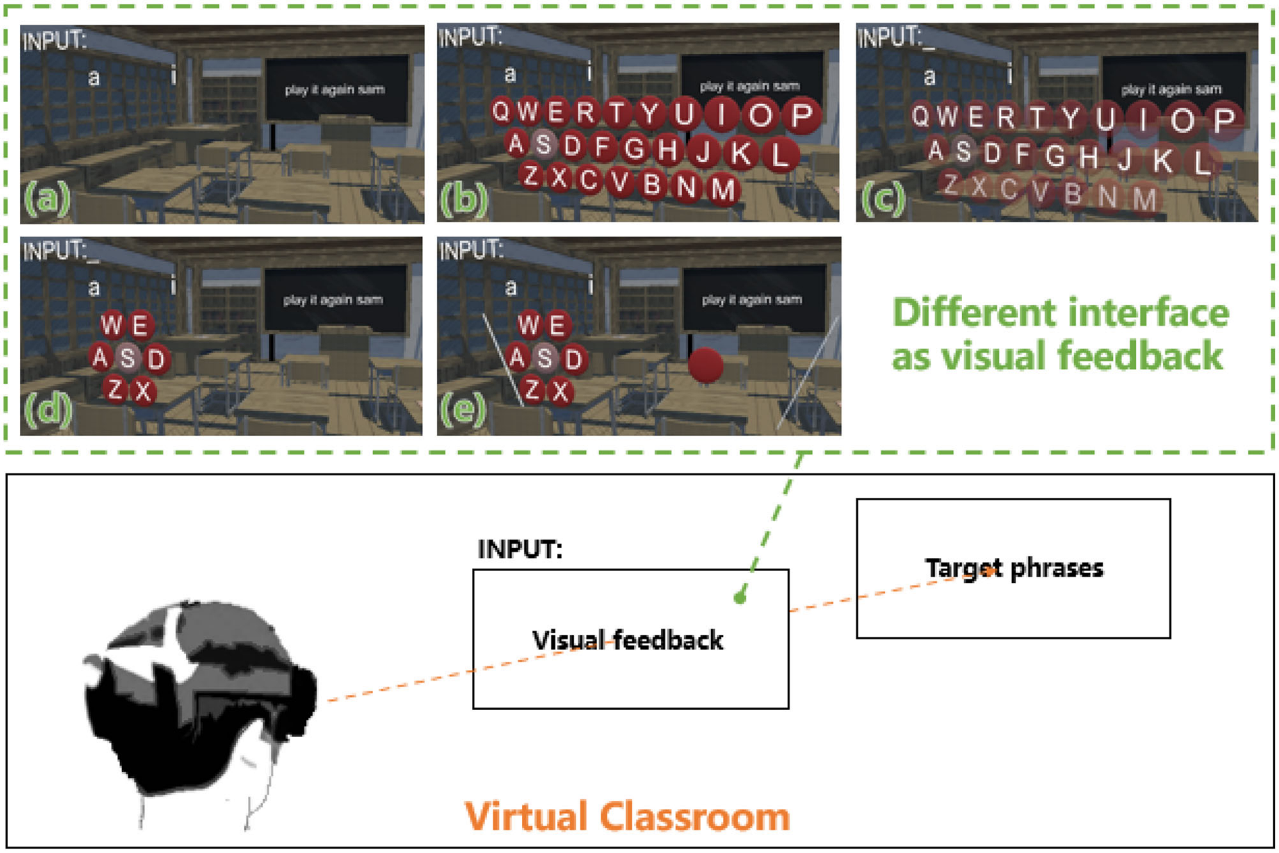


Figure 4. In this study, the user's initial line of sight, the visual feedback display area, and the target phrase are in a straight line. There are five different keyboard interfaces as visual feedback: (a) without visual feedback and no occlusion; (b) with a full QWERTY keyboard, which provides visual feedback but is fully occluded; (c) with a transparent full QWERTY keyboard, which provides visual feedback; (d) with a partial keyboard, which provides visual feedback; (e) adding spatial landmarks to (d).

transparent QWERTY keyboard with 50% transparency (Figure 4c). Session 4 is with our partial keyboard (Figure 4d). Session 5 adds the spatial landmarks to the visualization of Session 4 (Figure 4e). Before the whole study, we introduced how to enter text using our device, and participants were given 5 min to get familiar with the interaction of the keyboard. Before each session, participants were required to complete one phrase exercise. After each session, participants were required to fill out the NASA-TLX questionnaire (Hart, 2006). Each session took around 5 min for each participant. In total, 5 (sessions) × 10 (phrases) × 16 (participants) = 800 phrases of data were collected. The study has been approved by the Biology and Medical Ethics Committee of our university.

5.3.1.4. Metrics. We measure the speed of text entry by calculating the number of words per minute (WPM) (MacKenzie, 2013) with Equation 1:

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

Where T is the target phrase, and S is the time (in seconds) taken between the first and last press in each phrase.

The error rate of text input is measured by the total error rate (TER) with Equation 2 and the not corrected error rate (NCER) (Soukoreff & MacKenzie, 2003) with Equation 3:

$$NCER = \frac{INF + IF}{C + INF + IF} \times 100\% \tag{2}$$

$$TER = \frac{INF}{C + INF + IF} \times 100\% \tag{3}$$

Where INF represents the number of incorrect but uncorrected characters, which is calculated by the shortest edit distance between the input phrase and the target phrase; IF represents the number of incorrect but corrected characters, which is calculated by recording the number of times that the user pressed the delete key during the text entry process; and C represents the number of correct characters, which can be approximated as $C + INF = |T|$.

5.3.2. Results

We performed statistics on WPM, NCER, TER, and NASA-TLX scores for each session. The Shapiro-Wilk test was used to assess the normality of the data distribution. All the data conform to a normal distribution and are analyzed with ANOVA. We used Bonferroni correction in pair-wise comparisons and Greenhouse-Geisser adjustment in violations of the spherical hypothesis. The following are the results of the analysis.

5.3.2.1. Typing speed. No significant effect is found in the comparison of Session 1 vs Session 2 vs Session 3 ($F_{2,30} = 2.89, p = .0568, \eta_p^2 = .162$). Significant interaction effects are found in the comparison of Session 1 vs Session 4 ($F_{1,15} = 26.91, p = 3.81 \times 10^{-7}, \eta_p^2 = .642$) and Session 1 vs Session 5 ($F_{1,15} = 69.34, p = 2.49 \times 10^{-15}, \eta_p^2 = .822$), indicating that visual feedback can help users improve their typing efficiency. Significant interaction effects are found in the comparison of Session 2 vs Session 4 ($F_{1,15} = 18.28, p = 2.52 \times 10^{-5}, \eta_p^2 = .549$) and Session 2 vs Session 5 ($F_{1,15} = 57.19, p = 4.29 \times 10^{-13}, \eta_p^2 = .792$), indicating that our partial keyboard method in visual occlusion processing is superior to the full keyboard method. Significant interaction effects are also found in the comparison of Session 3 vs Session 4 ($F_{1,15} = 11.17, p = .0009, \eta_p^2 = .427$) and Session 3 vs Session 5 ($F_{1,15} = 45.01, p = 8.98 \times 10^{-11}, \eta_p^2 = .750$), indicating that our partial keyboard method is also superior to the transparent keyboard method. Significant interaction effects are found in the comparison of Session 4 vs Session 5 ($F_{1,15} = 8.56, p = .0037, \eta_p^2 = .363$), indicating that border and position indication have a significant effect on speed improvement.

Figure 5 shows the mean typing speed of the five sessions. Session 1 has the lowest mean typing speed of 20.98 WPM (s.e. = 5.30), while Session 5 has the fastest mean typing speed of 26.08 WPM (s.e. = 5.64), which is 24% faster. The difference in typing speed among Session 1, Session 2, and Session 3 is insignificant, and all are relatively low. Session 1 has no visual feedback, while Session 2 has the full keyboard for visual feedback but blocks the target phrase. Session 3 uses a transparent keyboard to reduce the occlusion of the target phrase, but the occlusion remains. The transparent keyboard may not be clear enough for some users. Typing speed was also improved in Session 5 compared to Session 4, where spatial landmarks can resolve the visual confusion caused by partial keyboard movement.

5.3.2.2. Error rate. For NCER, no significant interaction effect is found in the comparison of Session 1 vs Session 2 vs Session 3 vs Session 4 vs Session 5 ($F_{4,60} = 2.34, p = .054, \eta_p^2 = .135$), indicating that the NCER difference between the five Sessions was not significant. For TER, significant interaction effects are found in the comparison of Session 1 vs Session 2 ($F_{1,15} = 107.7, p = 6.35 \times 10^{-22}, \eta_p^2 = .878$), Session 1 vs Session 3 ($F_{1,15} = 113.5, p = 7.18 \times 10^{-23}, \eta_p^2 = .883$), and Session 1 vs Session 4 ($F_{1,15} = 30.08, p = 8.48 \times 10^{-8}, \eta_p^2 = .667$), indicating that the inclusion of visual feedback can significantly reduce the error rate. In addition, a significant interaction effect is found in the comparison of Session 4 vs Session 5 ($F_{1,15} = 5.35, p = .0213, \eta_p^2 = .263$), indicating that spatial landmarks can further reduce the error rate.

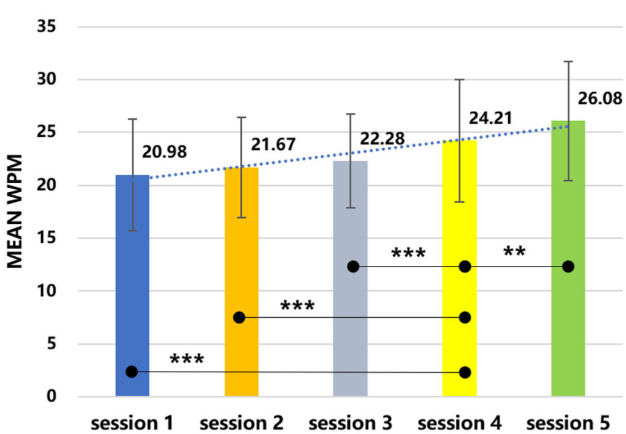


Figure 5. Mean WPM of the five sessions. Error bars indicate standard deviation. The dashed line indicates the trend of data change. Asterisks denote statistical significance between sessions.

Figure 6 shows the mean NCER and mean TER for the five sessions. The mean NCER and mean TER for Session 1 are both the highest at 1.59% (s.e. = 2.35%) and 9.00% (s.e. = 5.87%), while the mean NCER and mean TER for Session 3 are both the lowest at 0.98% (s.e. = 1.93%) and 3.37% (s.e. = 3.19%). In addition, the average NCER for sessions 1–5 were very similar and low, because we added SymSpell, whose word correction function helps reduce NCER. For TER, Session 3 with the transparent keyboard is the lowest, while Session 1 does not add any visual feedback, resulting in a high error rate. The error rate of the partial keyboard is somewhere in between.

5.3.2.3. Workload. Figure 7a shows the mean NASA-TLX workload scores for the five sessions. A significant effect is found in the comparison of Session 1 vs Session 2 ($F_{1,15} = 8.46, p = .0068, \eta_p^2 = .361$), indicating that the full keyboard method increases the workload compared to none-occlusion method, probably because the full keyboard obscures the target phrase to the user's discomfort, even though the full keyboard method is superior to none-occlusion method in terms of speed. No significant effect is found in the comparison of Session 2 vs Session 3 ($F_{1,15} = 2.39, p = .132, \eta_p^2 = .138$), indicating that transparent keyboards didn't give users a more favorable experience. The partial keyboard can significantly reduce the workload compared to none-occlusion (Session 1 vs Session 4: $p = 6.64 \times 10^{-5}$), full keyboard (Session 2 vs Session 4: $p = 8.85 \times 10^{-8}$) and transparent keyboard (Session 3 vs Session 4: $p = 1.86 \times 10^{-6}$). In addition, the inclusion of boundaries and position indication can also significantly reduce the workload (Session 4 vs Session 5: $p = .0093$).

Figure 7b shows the mean scores of each NASA-TLX dimension. Session 5 is the most preferred method in terms of all dimensions. For "Mental Demand" and "Performance," Session 1 is the least preferred method because it does not have any visual feedback, which makes users feel insecure when typing. For "Physical Demand," "Effort" and "Frustration," Session 2 is the least preferred method. For "Temporal Demand," Session 3 is the least preferred method. This may be due to the keyboard blocking the user's view, making users feel unpleasant.

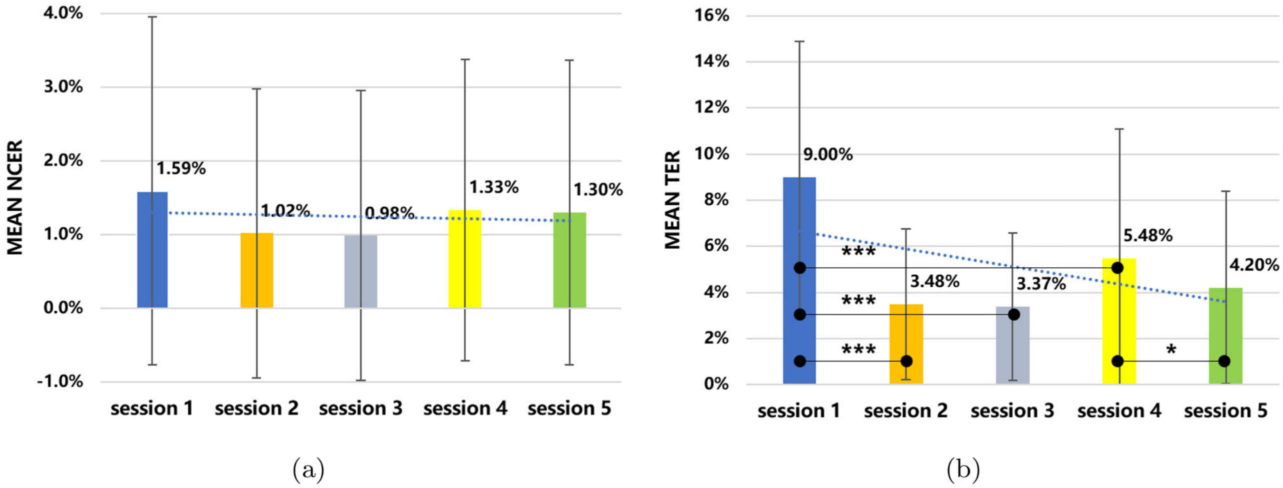


Figure 6. (a) Mean NCER, (b) mean TER of the five sessions. Error bars indicate standard deviation. The dashed line indicates the trend of data change. Asterisks denote statistical significance between sessions.

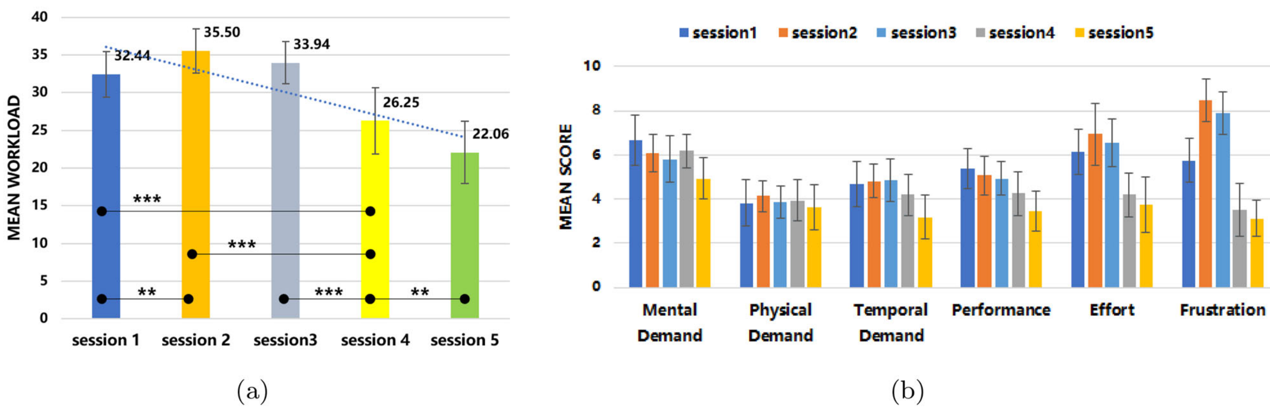


Figure 7. (a) Mean WORKLOAD of the five sessions; (b) mean score of each NASA-TLX dimension. The dashed line indicates the trend of data change. Asterisks denote statistical significance between sessions.

5.3.3. Discussion

From the above results, we find that text entry with none-occlusion does not obscure the vision, but its speed and accuracy are poor. On the other hand, the full keyboard gives visual feedback to the user, but it obscures the vision and has a low error rate but not a high speed. From the NASA-TLX results, it appears that these three methods lead to user dissatisfaction. Participants claimed that text entry without visual feedback gives the best visual experience, with no occlusion of the target phrase, but makes typing insecure. On the other hand, the full keyboard completely obscures the target phrase, which requires them to make frequent sight switches while typing. In addition, some users indicated that they still averted their eyes when performing Session 3 as they did in Session 2, and others indicated that the transparent keyboard was indeed much better than the opaque full keyboard, but the occlusion was still there. Previous study has also shown that transparent keyboards aren't as good as they're cracked up to be. In practice, it is not easy to get the right opacity for different background variations. The overlay design may also cause the main view and the keyboard view to interfere with each other (Sun et al., 2019).

Compared to the previous three methods, our partial keyboard is a compromise. It not only provides the user with proper visual feedback, but also ensures that it does not completely obscure the user's view. Regarding typing speed, our partial keyboard method is far superior to the above three methods. The error rate is similar to the full keyboard method. Participants said the moving partial keyboard was interesting to them, and the typing experience was better than the first three sessions. When it comes to the difference in experience between Session 4 and Session 5, participants claimed since the keyboard boundaries were not visible, the method in Session 4 would cause confusion. After adding simple dots and lines for indication, this confusion was almost eliminated.

6. Underlying keyboard optimization

In this section, we present an optimization algorithm for the underlying keyboard that performs intra-block optimization

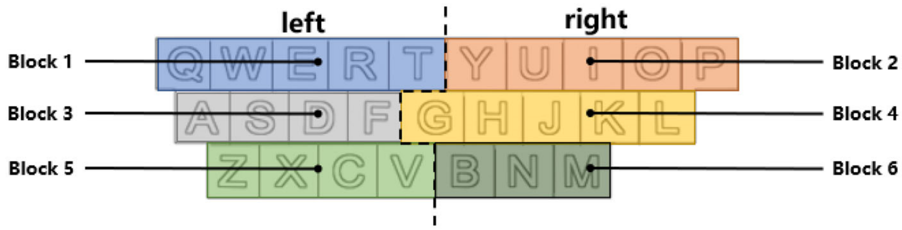


Figure 8. The standard QWERTY keyboard is divided into six blocks.

based on users' eyes-free typing data. We designed a pilot user study to evaluate the performance of our algorithm.

6.1. Division of keyboard blocks

Since our method is based on two-thumb typing, it inevitably involves the left-hand and right-hand areas of the keyboard. Generally, QWERT/ASDF/ZXCV is in the left-hand area; YUIOP/GHJKL/BNM is in the right-hand area. We then split the left-hand and right-hand areas separately according to the keyboard line, so it forms six keyboard blocks, as shown in Figure 8. According to the definition of keyboard blocks, we adopt the keyboard optimization strategy of "intra-block modification," which means for each block, the position and size are fixed. The size of the keys within each block is determined by the ratio of the evaluation values within the block. This is done so as not to destroy the relative positions of the keys. As mentioned earlier, the eyes-free input relies heavily on the muscle memory of the fingers for the keyboard. If the relative positions of the keys are destroyed too severely, it inevitably leads to a decrease in input efficiency and an increase in the error rate.

6.2. Intra-block optimization algorithm

We propose a keyboard optimization algorithm to improve the typing speed and accuracy. Based on the collected typing data, we first calculate the center point offset, which indicates how much each key is offset relative to the standard layout of the QWERTY keyboard on the X-axis. As well as using the 95% confidence ellipse for each key to calculate the error touching rate, which indicates how much each key is interfered with by other keys. We then use the Entropy Weighting Method (Y. Zhu et al., 2020) to obtain the weights of the above two parameters and obtain the weighted evaluation value of each key. Based on the evaluation value of each key, we scale the size of each key within the block proportionally to get the width.

The underlying keyboard optimization algorithm is shown in Algorithm Algorithm 1. Given keys of the keyboard K, blocks of the keyboard B, typing data X_t, standard key data X, key width W, optimized key width W_o is calculated. Since the user is holding the phone horizontally during text input, we take the horizontal as the x-axis and the vertical as the y-axis, and the offset of the touch point on the y-axis is minimal and almost negligible. Therefore, we only considered the data on the X-axis when designing the algorithm.

Algorithm 1 Underlying keyboard optimization algorithm

Input: keys of the keyboard K, blocks of the keyboard B, typing data X_t, standard key data X, key width W
Output: optimized key width W_o

1: **for** $k_i \in K$ **do** $\Delta X_i \leftarrow Mean(X_{t_i}) - X_i$
2: **end for**
3: **for** $k_i k_j \in K$ñK **do**
4: **if** $isIntersect(k_i, k_j)$ **then** $s_{ij} \leftarrow area_{ij} * (sum_{ij}/n), S_i \leftarrow S_i + s_{ij}$
5: **end if**
6: **end for**
7: $\{w_1, w_2\} \leftarrow EWM(normal(\Delta X_i), normal(S_i))$
8: **for** $k_i \in K$ **do** $F_i \leftarrow w_1 * normal(\Delta X_i) + w_2 * normal(S_i)$
9: **end for**
10: **for** $b_j \in B$ **do** $F_{b_j} \leftarrow sum(F, j), W_{b_j} \leftarrow sum(W, j)$
11: **end for**
12: **for** $k_i b_j \in K$ñB **do**
13: **if** $k_i \in b_j$ **then** $W_{o_i} \leftarrow W_{b_j} * (F_i / F_{b_j})$
14: **end if**
15: **end for**
16: **return** W_o

In Algorithm 1, we first calculate the average value of the collected typing data for each key on the X-axis $Mean(X_{t_i})$, and then subtract the position of each key on the standard QWERTY keyboard layout X_i to obtain the centroid offset ΔX_i (line 1). We then sum up the approximate intersection area of the 95% confidence ellipse between each key and the other keys as the error touching rate S_i (lines 2–3), where *isIntersect* indicates that there is an intersection between the touch ranges of two keys k_i and k_j, $area_{ij}$ denotes the area of the smallest rectangle that can cover the confidence ellipses of k_i and k_j, sum_{ij}/n approximately indicates the percentage of intersection area in $area_{ij}$. To do this, we cast $n(n = 1,000,000)$ points randomly within $area_{ij}$, and sum_{ij} denotes the number of points in the intersection area, and the points in the intersection area should satisfy Equation 4.

$$\begin{cases} \begin{bmatrix} X'_{random_i} \\ Y'_{random_i} \end{bmatrix} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{bmatrix}^{-1} \times \begin{bmatrix} X_{random_i} \\ Y_{random_i} \end{bmatrix} \\ \begin{bmatrix} X'_{random_j} \\ Y'_{random_j} \end{bmatrix} = \begin{bmatrix} \cos\theta_j & -\sin\theta_j \\ \sin\theta_j & \cos\theta_j \end{bmatrix}^{-1} \times \begin{bmatrix} X_{random_j} \\ Y_{random_j} \end{bmatrix} \\ {X'_{random_i}}^2 / a_i^2 + {Y'_{random_i}}^2 / b_i^2 \le 1 \\ {X'_{random_j}}^2 / a_j^2 + {Y'_{random_j}}^2 / b_j^2 \le 1 \end{cases} \tag{4}$$

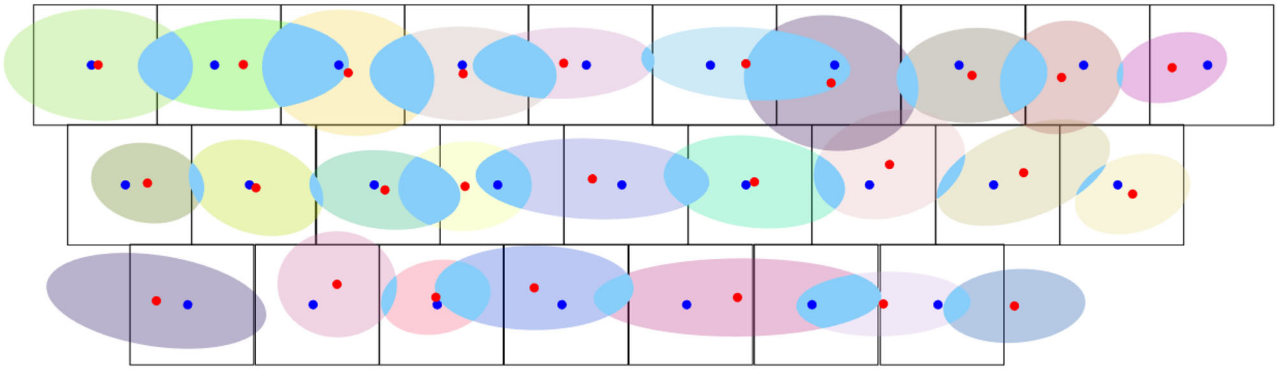


Figure 9. 95% confidence ellipse of user touch points.

Table 1. The width ratio of keys in each block.

Blocks	Width ratio of keys				
block1	Q-0.75	W-1.25	E-1.20	R-1.00	T-0.80
block2	Y-1.35	U-1.10	I-0.85	O-0.90	P-0.80
block3	A-0.85	S-0.70	D-1.10	F-1.35	–
block4	G-1.40	H-0.90	J-0.95	K-1.00	–
block5	Z-0.90	X-0.80	C-0.80	V-1.50	–
block6	B-1.00	N-1.15	M-0.85	–	–

where θ is the offset angle of the confidence ellipse, a and b are the lengths of the long and short axes of the ellipse, respectively. X_{random} and Y_{random} are the locations of randomly cast points within the range of $area_{ij}$. X'_{random} and Y'_{random} are the values of X_{random} and Y_{random} in the corresponding elliptical coordinate system.

Next, we calculate the weights of center point offset w_1 and error touching rate w_2 by using the Entropy Weighting Method (line 4). Based on these weights, we calculate the evaluation value F_i of each key (line 5) and the sum of evaluation values and widths of each block (line 6), where $sum\ (F, j)$ denotes the total evaluation value F_{b_j} of the block b_j and $sum\ (W, j)$ denotes the total width W_{b_j} of the block b_j. Finally, we redistribute the width of each key W_{o_i} proportionally within the block b_j (lines 7–9).

6.3. Data collection and optimization results

6.3.1. Eyes-free typing data collection

This study aims to obtain users' eyes-free typing data for further analysis. Many previous studies (Leng et al., 2022; Y. Lu et al., 2017; S. Zhu et al., 2019) have analyzed users' eyes-free typing data, but none of the current work addresses two-thumb eyes-free typing on touchscreen.

6.3.1.1. Participants. Ten participants (seven males and three females, aged 21–25) from our university were recruited for this study. They are all familiar with the basic structure of the QWERTY keyboard.

6.3.1.2. Hardware setup. A Redmi Note11 running Andriod 11 is used to run the input program. The screen size of the mobile phone is 6.6-inch and measures 163.56 × 75.78 mm. A Honor MagicBook 2019 laptop with an AMD RYZEN 7 processor and AMD Radeon(TM) RX Vega 10Graphics graphics card is used to collect the data. The screen size of the laptop is 14 inches, and the resolution is 1920*1080. The programs are developed in C# in Unity 2021.3.

6.3.1.3. Task and procedure. We use a Wizard of Oz testing (Dahlbäck et al., 1993). The users do not have to care whether they hit the right key, and we always give the correct output. The user was asked to sit about 0.5 m away from the laptop screen to type using the mobile phone, and we asked the user to type using two thumbs and try to focus on the computer screen rather than the phone while typing. The task was to enter 10 phrases, which were randomly generated from the Mackenzie phrase set (MacKenzie & Soukoreff, 2003).

6.3.1.4. Data analysis. We collected typing data for 100 phrases. For each key, we removed outliers that were more than three times the standard deviation away from the collected centroid in either X or Y dimension. The result is as follows:

Figure 9 shows the distribution of the touch points collected in this study. The ellipse corresponding to each button covers it with 95% confidence. The blue dot is the center of each key, while the red dot is the center of the 95% confidence ellipse for each key. The light-blue portion indicates the overlap between the keys. It is easy to see that the touch points are pretty chaotic, with considerable overlap between keys. However, the layout of the 95% confidence ellipse formed by the touch points is essentially the same as the layout of the QWERTY keyboard key center points, and the average center point offset is calculated to be about 2.1 cm on the X-axis and about 0.3 cm on the Y-axis. That is, the center point offsets are small overall, indicating that users can transfer the spatial and muscle memory they have developed on the visible QWERTY keyboard to the eyes-free input.

In summary, the offset of the touch points demonstrates the feasibility of users performing eyes-free typing on a smooth touchscreen surface. However, the key-to-key overlap also suggests the need for reasonable visual feedback for correction and optimization of the underlying keyboard.

6.3.2. Optimization results

Using the eyes-free typing data and processing it with the algorithm of subsection 6.2, we obtain the final optimization results. Table 1 shows the width ratio of each key after the algorithm optimization (the size of the key before modification is 1). Figure 10 visualizes the variation of the key widths within the block. It is easy to see that each block follows the rule that keys near the center of the keyboard are larger and keys near the edge of the keyboard are smaller. After expanding the A, L, Z, and M keys to the edge of the screen, it will eventually appear: it is larger on both sides and smaller in the middle for each block. When we are ready to type, the natural position of our thumbs is in the middle of the block, and when we want to type, the keys in the middle of each block are easier to touch, while the keys in the positions on the sides are harder to touch. Thus, the "big-small-big" arrangement fits our typing habits.

6.4. Pilot user study 2: Evaluate underlying keyboard optimization algorithm

In this section, we design a user study to evaluate the performance of our underlying keyboard optimization. We compare the keyboard optimized by our algorithm with the keyboard optimized by the absolute algorithm of Blindtype (Y. Lu et al., 2017) and the standard keyboard. In addition, we compare the performance of these different underlying keyboards in two cases: with the partial keyboard as visual feedback and without any visual feedback. We use this study to prove that our algorithm can improve users' typing efficiency.

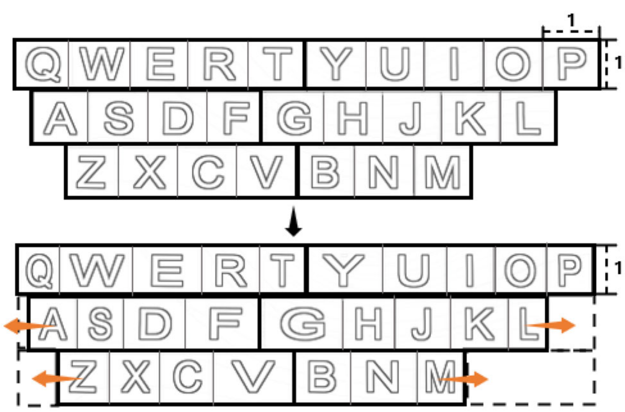


Figure 10. The underlying keyboard model after key-width optimization.

6.4.1. User study design

6.4.1.1. Participants, hardware setup and metrics. The same 16 participants from pilot user study 1 participated in this study. The hardware setup and test methodology of this study were also the same as that of pilot user study 1.

6.4.1.2. Task and procedure. The experiment used a 2×3 within-subjects design, i.e. all participants were required to attend six sessions. The task was to enter 10 phrases in each session. The order of the conditions was counterbalanced across participants, and phrases in each session were randomly generated from the Mackenzie phrase set (MacKenzie & Soukoreff, 2003). Participants were asked to enter the text "quickly and accurately." There are two conditions we need to consider. Condition 1 has two options: with and without visual feedback. Condition 2 has three options: with the standard QWERTY underlying keyboard, with the underlying keyboard optimized by Blindtype's algorithm, and with the underlying keyboard optimized by our algorithm. We label the sessions according to different combinations of conditions. Sessions 1,3,5 are without any visual feedback (shown in Figure 11a,c,e), and Sessions 2,4,6 are with partial keyboard for visual feedback (shown in Figure 11b,d,f). Session 1 and 2 are with standard QWERTY as underlying keyboard (shown in Figure 11a,b); Sessions 3 and 4 are with the keyboard optimized by Blindtype's algorithm (shown in Figure 11c,d); Sessions 5 and 6 are with the keyboard optimized by our algorithm (shown in Figure 11e,f). Each session took around 5 min for each participant. Data for a total of 2(visual feedback) $\times$3(underlying keyboard) $\times$10(phrases) $\times$16(participants) $=$ 960 phrases were collected. The study

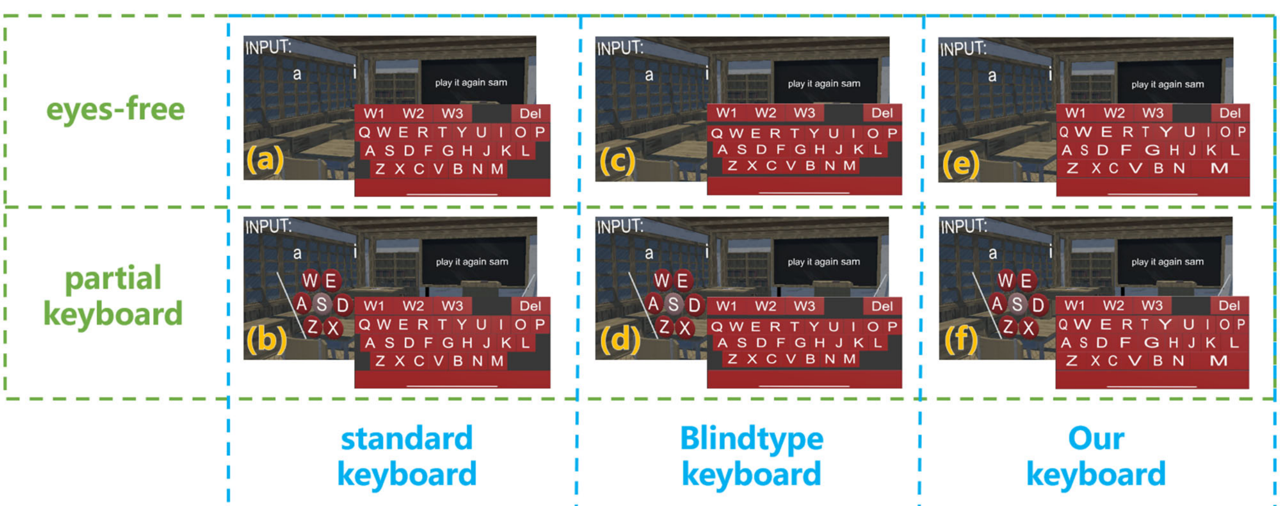


Figure 11. Different combinations of conditions for the six sessions. The top left corner of each Sub-figure shows the display interface, which is divided into "eyes-free" and "partial keyboard," and the top right corner shows the underlying keyboard interface, which is divided into "standard keyboard," "Blindtype's keyboard" and "our keyboard.".

has been approved by Biology and Medical Ethics Committee of our university.

6.4.2. Results

We performed statistics on WPM, NCER, and TER for each session. The Shapiro-Wilk test was used to assess the normality of the data distribution. All the data conform to a normal distribution and are analyzed with ANOVA. We used Bonferroni correction in pair-wise comparisons and Greenhouse-Geisser adjustment in violations of the spherical hypothesis. The following are the results of the analysis.

6.4.2.1. Typing speed

The ANOVA results showed that "visual feedback"$(F_{1,15} = 42.91, p = 3.51 \times 10^{-10}, \eta_p^2 = .741)$ and "underlying keyboard" $(F_{2,30} = 5.1, p = .0068, \eta_p^2 = .254)$ have a significant effect on the typing speed, indicating that the addition of the partial keyboard and the underlying keyboard optimization algorithm has a significant effect on the typing speed. Significant interaction effects are found in Session 2 vs Session 6 $(F_{1,15} = 11.65, p = .001, \eta_p^2 = .437)$, and Session 4 vs Session 6 $(F_{1,15} = 4.44, p = .0383, \eta_p^2 = .228)$, indicating that our optimization algorithm improves significantly with partial keyboard. The two methods complement each other. In contrast, no significant effects are found in the comparison of Session 1 vs Session 5 $(F_{1,15} = 2.31, p = .132, \eta_p^2 = .133)$ and Session 3 vs Session 5 $(F_{1,15} = .98, p = .324, \eta_p^2 = .061)$, shows that the algorithm is not very effective with no visual feedback.

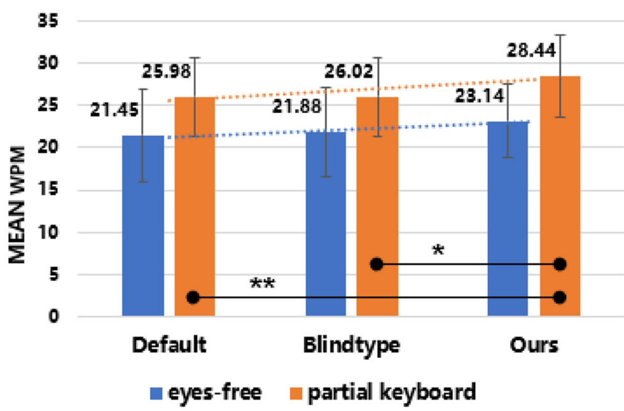


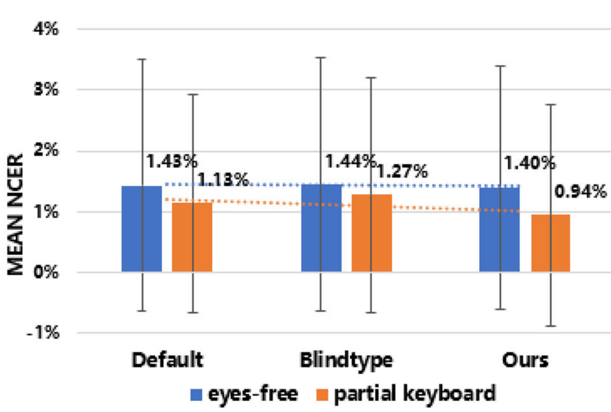
Figure 12. Mean WPM of the six sessions. Error bars indicate standard deviation. The dashed line indicates the trend of data change. Asterisks denote statistical significance between sessions.

Figure 12 shows the mean typing speed for the six sessions. Session 1 has the lowest mean typing speed of 21.45 WPM (s.e. = 5.5), while Session 6 has the fastest mean typing speed of 28.44 WPM (s.e. = 4.91). The mean typing speed of Session 1 vs Session 3 and Session 2 vs Session 4 is almost the same, which indicates that Blindtype's keyboard optimization algorithm may not apply to the two-thumb typing method, and the optimized keyboard does not differ much from the default keyboard. The typing speed of Session 5 is the fastest in Sessions 1, 3, and 5, and the typing speed of Session 6 is the highest in Sessions 2, 4, 6. In particular, Session 6 was significantly higher than Session 4 and Session 2, indicating that our algorithm can improve typing speed with the partial keyboard.

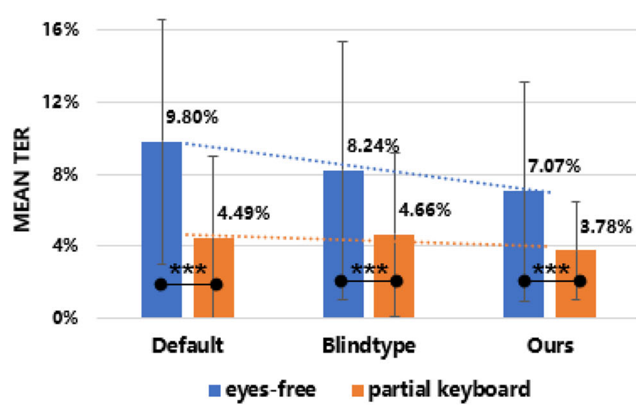
6.4.2.2. Error rate

For NCER, no significant interaction effects are found in both "visual feedback" $(F_{1,15} = 1.54, p = .2153, \eta_p^2 = .093)$ and "underlying keyboard" $(F_{2,30} = .18, p = .833, \eta_p^2 = .012)$. Indicating that the NCER has little relationship with the visual feedback. For TER, "visual feedback" $(F_{1,15} = 32.47, p = 3.56 \times 10^{-8}, \eta_p^2 = .684)$ has a significant effect, while "underlying keyboard" $(F_{2,30} = 1.74, p = .178, \eta_p^2 = .104)$ has no significant effect. Meanwhile, no significant interaction effects are found in Session 1 vs Session 3 vs Session 5 $(F_{2,30} = 1.68, p = .191, \eta_p^2 = .101)$ and Session 2 vs Session 4 vs Session 6 $(F_{2,30} = .54, p = .584, \eta_p^2 = .035)$. This indicates that the main factor influencing the error rate is the visual feedback and is less related to the underlying keyboard.

Figure 13 shows the mean NCER and TER for the six sessions. For NCER, they are all very low due to the inclusion of word prediction and correction in each session. For TER, including the partial keyboard for visual feedback significantly reduced the error rate. In addition, the difference



(a)



(b)

Figure 13. (a) Mean NCER and (b) mean TER of the six sessions. Error bars indicate standard deviation. The dashed line indicates the trend of data change. Asterisks denote statistical significance between sessions.

between the default keyboard and the Blindtype algorithm-optimized keyboard is insignificant. And the error rate of our algorithm-optimized keyboard is very low, only 3.78% (s.e. = 2.68%).

6.4.3. Discussion

Our underlying keyboard optimization algorithm can improve efficiency and accuracy with or without visual feedback. In addition, Session 2 shows a 21% increase in typing speed compared to Session 1. In contrast, Session 5 shows an 8% increase in typing speed compared to Session 1. The partially visible keyboard is the root cause of the increase in typing speed. This is because the partially visible keyboard provides the user with visual feedback while increasing the user's confidence when typing, and optimizing the underlying keyboard still doesn't address the central issue of visual feedback. However, the underlying keyboard optimization algorithms still help with typing speed improvement because it reduce the probability of keystroke errors and make typing more comfortable for the user. Session 6 shows a 33% increase in typing speed compared to Session 1. Two optimizations can help users increase typing speed at the same time. Our partial keyboard and optimization algorithm work well together.

From the above results, it is easy to see that Blindtype's absolute algorithm does not perform well for two-thumb typing. When using Blindtype's algorithm to optimize the underlying keyboard, the optimized keyboard does not differ much from the default keyboard. This is because the Blindtype algorithm scales the overall keyboard only based on the centroid offset of each key, and it is a one-thumb typing method which does not fully consider the key blocks of the keyboard. In contrast, our algorithm combines the three factors: centroid offset, error touching rate, and key blocks, which obtain a significant performance improvement. Many participants claimed it was easier to type in Session 5 and Session 6. Especially during fast typing, there are significantly fewer cases of wrong touching.

7. User study: Evaluate performance and learnability in the MR environment

Our approach supports both VR and MR. Pilot user study 1 and pilot user study 2 were conducted in a VR environment (virtual classroom). In this user study, we focus on MR scenarios. We conduct a 5-day user study to evaluate the performance and learnability of our Partially Visible Keyboard and explore how the performance of two groups, the novice group and the potential expert group, would improve in a 5-day practice. We use the same hardware setup as the pilot user study.

7.1. User study design

7.1.1. Participants

Eight participants (five males and three females, aged 21–25) were recruited to conduct the user study. We divided them into two groups: a novice group and a potential expert group. None of the participants in the novice group have participated in the pilot user study, and none of them have experienced text entry in MR. The participants in the expert group were from the pilot user study. We ranked all participants from the pilot user study based on their performance, selected the top four best performers, and invited them to continue participating in the 5-day study to form the potential expert group.

7.1.2. Task and procedure

The entire study consisted of five sessions, one for each day. For all users, the task was to enter 10 phrases in the MR environment, and phrases in each session were randomly generated from the Mackenzie phrase set (MacKenzie & Soukoreff, 2003). Participants were asked to enter the text "quickly and accurately." For novice users, We showed them how to use our method for input and gave them 5 min to try it out before the session started. Each session took around 5 min for each participant. Data for a total of 4 (participants) × 2 (groups) × 5 (sessions) × 10 (phrases) = 400 phrases were collected. The study has been approved by the Biology and Medical Ethics Committee of our university.

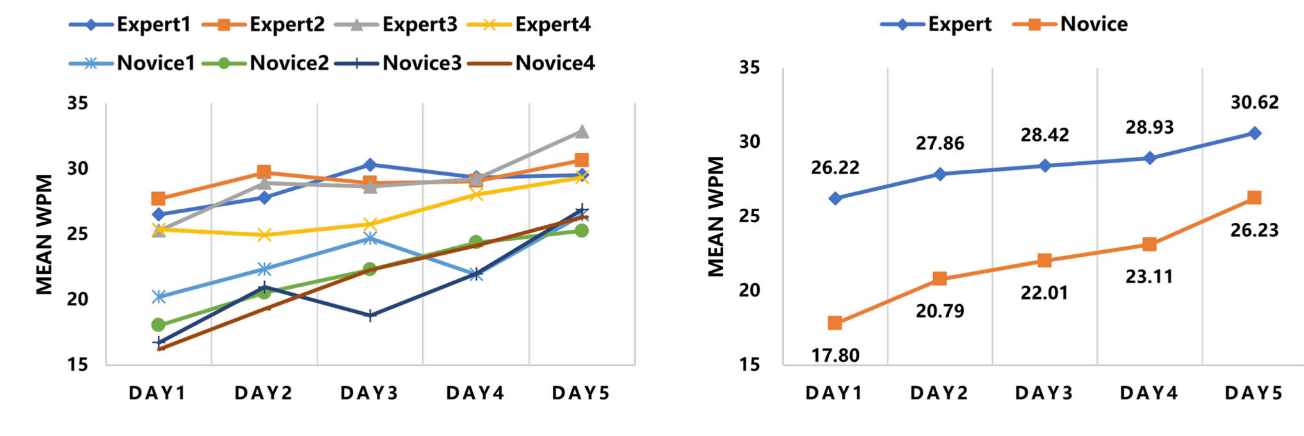
7.2. Results

We performed statistics on WPM, NCER, and TER for each session. The Shapiro-Wilk test was used to assess the normality of the data distribution. All of the data conform to a normal distribution. We used ANOVA for analysis, with "session" (1–5) as the within-subjects variable and "group" (novice and potential expert) as the between-subjects variable. We used Bonferroni correction in pair-wise comparisons and Greenhouse-Geisser adjustment in violations of the spherical hypothesis. The following are the results of the analysis.

7.2.1. Typing speed

The ANOVA results show that "group" ($F_{1,4} = 17.01, p = .0033, \eta_p^2 = .809$) and "session" ($F_{4,28} = 3.05, p = .0296, \eta_p^2 = .303$) both have a significant effect on typing speed. There is a significant difference in typing speed within five days, with the potential expert group typing at a significantly higher speed than the novice group. Within the potential expert group, significant interaction effects are found only for sessions 1vs4 and 1vs5 (both $p < .05$). Within the novice group, significant interaction effects are found for sessions 1vs2, 1vs3, 1vs4, 1vs5, 2vs4, 2vs5, 3vs5, and 4vs5 (all $p < .05$). This indicates that the two groups have significantly different learning effects, with the novice group showing a much greater increase in typing speed than the potential expert group. After five days of training, the learning curve of the novice group was still increasing.

Figure 14a shows the mean typing speed of the 8 participants. The mean speed of the 8 participants for all sessions over five days is 25.20 WPM. The average speed of all participants is 22.01 WPM (s.e. = 4.71) in the first test and



(a) (b)

Figure 14. (a) Mean typing speed of 8 participants and (b) mean typing speed of novice group and potential expert group of 5 days.

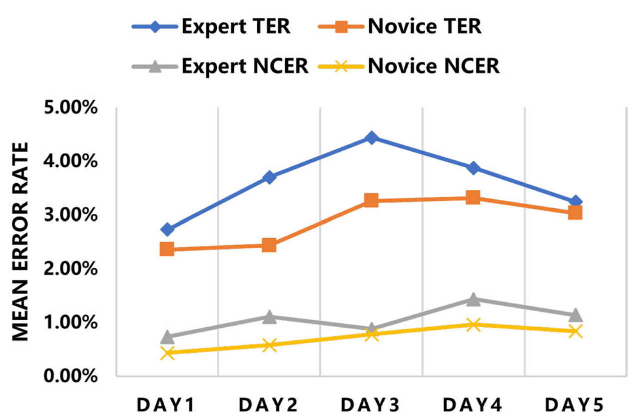


Figure 15. Mean NCER and TER of novice group and potential expert group of 5 days.

reached 28.42 WPM (s.e. = 2.61) in the last test, an increase of 29.12%. The fastest speed is reached by the participant from the potential expert group in the fifth session, reaching 32.88 WPM, while its typing speed on the first day is only 25.28 WPM, which is the lowest among the potential expert group. Figure 14b shows the average typing speed of the two groups in each session. The average speed of the novice group is 21.99 WPM (s.e. = 3.09), while the average speed of the potential expert group is 28.41 WPM (s.e. = 1.60). The speed of the potential expert group increased from 26.22 WPM (s.e. = 1.14) in the first session to 30.62 WPM (s.e. = 1.61) in the last session, an increase of 16.78%. And the speed of the novice group increase from 17.80 WPM (s.e. = 1.79) in the first session to 26.23 WPM (s.e. = 0.69), an increase of 47.36%.

7.2.2. Error rate

For NCER, no significant effects are found in both "group"($F_{1,4} = 4.96, p = .0566, \eta_p^2 = .554$) and "session" ($F_{4,28} = 1.48, p = .334, \eta_p^2 = .175$). For TER, no significant effects are found in "group" ($F_{1,4} = 4.17, p = .0754, \eta_p^2 = .510$) and "session" ($F_{4,28} = 1.45, p = .343, \eta_p^2 = .172$), indicating that the number of training days does not have a significant effect on error rate.

Figure 15 shows the mean NCER and TER for the novice group and the potential expert group for five sessions. The mean NCER for all sessions over five days for the 8 participants is 0.89% and the mean TER is 3.24%. For the mean NCER, it is 0.72% (s.e. = 0.21%) for the novice group and 1.06% (s.e. = 0.27%) for the potential expert group. For the mean TER, it is 2.88% (s.e. = 0.45%) for the novice group and 3.60% (s.e. = 0.64%) for the potential expert group. We also find that both TER and NCER of the potential expert group are higher than those of the novice group. This may be due to their faster typing speed, which increases the probability of typing errors. Moreover, as the typing speed of the novice group users increases, their error rate also increases slightly.

7.3. Discussion

We compare our Partially Visible Keyboard with the state-of-the-art touchscreen-based input methods i's Free (S. Zhu et al., 2019) and Blindtype (Y. Lu et al., 2017).

In terms of typing speed, novice users can reach a speed of 26.23 WPM, while potential users can reach 30.62 WPM. i's Free and Blindtype are also mobile phone touchscreen-based methods, with a maximum typing speed of 23.27 WPM and 22.77 WPM, respectively. Our method is about 25% faster than theirs. Our partial keyboard provides users with reasonable visual feedback, and the keyboard optimization algorithm allows users to type more naturally, which is the key to our high typing speed.

In terms of accuracy, the NCER and TER of our method are low, 0.72% and 2.88% for novice users and 1.06% and 3.60% for potential expert users, respectively. Blindtype's personalized relative algorithm has an NCER of 2.19%, a CER of 5.96%, and a PER (prediction error rate) of 2.26%. And the error rate of i's Free is 2.14%. The error rate of our method is significantly lower than Blindtype and similar to i's Free. For accuracy, our method is not inferior to the most advanced methods either.

In terms of learnability, after 40 phrases of training, the speed of novice users increased by 47.36% and that of potential expert users increased by 16.78%, which indicates that our text entry method is very user-friendly and easy to learn. We use a typical mobile phone as the input device and choose the most familiar two-thumb input as the input method.

8. Conclusion, limitations, and future work

In this paper, we have proposed a light-occlusion text entry method, a partial keyboard-based eyes-free text entry method. We first designed a partial keyboard interface for visual feedback. Then, we analyze the users' eyes-free typing data and propose an underlying keyboard optimization algorithm. Compared to the state-of-the-art touchscreen-based method, our method is faster and less occluded in MR environments. After five training days, novice users reach an average typing speed of 26.23 WPM, and expert users reach 30.62 WPM. The highest typing speed could reach 32.88 WPM.

Although our method has proven efficient and accurate, it has certain limitations. First, in some characteristic environments, our partial keyboard may obscure the target phrase and not achieve "none-occlusion." Second, our underlying keyboard does not change once it is determined, while previous research (Shi et al., 2018) shows that the keyboard position and size should be changed over time, and the text input habits of different users vary somewhat.

In the future, we will improve the following aspects: First, we can use multi-sensory channel fusion interactions to remove occlusion. Second, we will use personalization data to improve the underlying keyboard optimization algorithm so that the underlying keyboard can automatically adapt to the user's habits.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work is supported by the National Natural Science Foundation of China through Project [61932003, 62372026], by Beijing Science and Technology Plan Project [Z221100007722004], and by National Key R&D plan [2019YFC1521102].

References

Adhikary, J., & Vertanen, K. (2021). Text entry in virtual environments using speech and a midair keyboard. *IEEE Transactions on Visualization and Computer Graphics*, *27*(5), 2648–2658. https://doi.org/10.1109/TVCG.2021.3067776

Azenkot, S., Wobbrock, J. O., Prasain, S., Ladner, R. E. (2012). *Input finger detection for nonvisual touch screen text entry in perkinput* [Paper presentation]. Proceedings of Graphics Interface 2012 (pp. 121–129).

Azenkot, S., & Zhai, S. (2012). *Touch behavior with different postures on soft smartphone keyboards* [Paper presentation]. Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services (pp. 251–260). https://doi.org/10.1145/2371574.2371612

Bi, X., Chelba, C., Ouyang, T., Partridge, K., & Zhai, S. (2012). *Bimanual gesture keyboard* [Paper presentation]. Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (pp. 137–146). https://doi.org/10.1145/2380116.2380136

Boletsis, C., & Kongsvik, S. (2019). Text input in virtual reality: A preliminary evaluation of the drum-like vr keyboard. *Technologies*, *7*(2), 31. https://doi.org/10.3390/technologies7020031

Dahlbäck, N., Jönsson, A., & Ahrenberg, L. (1993). *Wizard of oz studies: Why and how* [Paper presentation]. Proceedings of the 1st International Conference on Intelligent User Interfaces (pp. 193–200). https://doi.org/10.1145/169891.169968

Garbe, W. (2019). *Symspell*. https://github.com/wolfgarbe/SymSpell

Hart, S. G. (2006). Nasa-task load index (nasa-tlx); 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *50*(9), 904–908. https://doi.org/10.1177/154193120605000909

Jiang, H., & Weng, D. (2020). *Hipad: Text entry for head-mounted displays using circular touchpad* [Paper presentation]. 2020 IEEE Conference on Virtual Reality and 3d User Interfaces (VR) (pp. 692–703). https://doi.org/10.1109/VR46266.2020.00092

Leng, J., Wang, L., Liu, X., Shi, X., & Wang, M. (2022). Efficient flower text entry in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, *28*(11), 3662–3672. https://doi.org/10.1109/TVCG.2022.3203101

Lu, X., Yu, D., Liang, H.-N., & Goncalves, J. (2021). *itext: Hands-free text entry on an imaginary keyboard for augmented reality systems* [Paper presentation]. The 34th Annual ACM Symposium on User Interface Software and Technology (pp. 815–825). https://doi.org/10.1145/3472749.3474788

Lu, Y., Yu, C., Fan, S., Bi, X., & Shi, Y. (2019). *Typing on split keyboards with peripheral vision* [Paper presentation]. Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (pp. 1–12). https://doi.org/10.1145/3290605.3300430

Lu, Y., Yu, C., Yi, X., Shi, Y., & Zhao, S. (2017). Blindtype: Eyes-free text entry on handheld touchpad by leveraging thumb's muscle memory. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *1*(2), 1–24. https://doi.org/10.1145/3090083

Lystbæk, M. N., Pfeuffer, K., Grønbæk, J. E. S., & Gellersen, H. (2022). Exploring gaze for assisting freehand selection-based text entry in ar. *Proceedings of the ACM on Human-Computer Interaction*, *6*(ETRA), 1–16. https://doi.org/10.1145/3530882

MacKenzie, I. S. (2002). *KSPC (keystrokes per character) as a characteristic of text entry techniques* [Paper presentation]. Proceedings of the Human Computer Interaction with Mobile Devices: 4th International Symposium, Mobile HCI 2002 Pisa, Italy (pp. 195–210).

MacKenzie, I. S. (2018). Fitts' law. In *The Wiley handbook of human computer interaction* (Vol. 1, Chapter 17, pp. 347–370). Wiley. https://doi.org/10.1002/9781118976005.ch17

MacKenzie, I. S. (2013). A note on the validity of the Shannon formulation for Fitts' index of difficulty. *Open Journal of Applied Sciences*, *03*(06), 360–368. https://doi.org/10.4236/ojapps.2013.36046

MacKenzie, I. S., & Soukoreff, R. W. (2003). *Phrase sets for evaluating text entry techniques* [Paper presentation]. Chi'03 Extended Abstracts on Human Factors in Computing Systems (pp. 754–755). https://doi.org/10.1145/765891.765971

MacKenzie, I. S., Zhang, S. X., & Soukoreff, R. W. (1999). Text entry using soft keyboards. *Behaviour & Information Technology*, *18*(4), 235–244. https://doi.org/10.1080/014492999118995

Pick, S., Puika, A. S., & Kuhlen, T. W. (2016). *Swifter: Design and evaluation of a speech-based text input metaphor for immersive virtual environments* [Paper presentation]. 2016 IEEE Symposium on 3d User Interfaces (3dui) (pp. 109–112). https://doi.org/10.1109/3DUI.2016.7460039

Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *26*(1), 43–49. https://doi.org/10.1109/TASSP.1978.1163055

Shi, W., Yu, C., Yi, X., Li, Z., & Shi, Y. (2018). Toast: Ten-finger eyes-free typing on touchable surfaces. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *2*(1), 1–23. https://doi.org/10.1145/3191765

Shneiderman, B. (2000). The limits of speech recognition. *Communications of the ACM*, *43*(9), 63–65. https://doi.org/10.1145/348941.348990

Soukoreff, R. W., & MacKenzie, I. S. (2003). *Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric* [Paper presentation]. Proceedings of the Sigchi Conference on Human Factors in Computing Systems (pp. 113–120).

Southern, C., Clawson, J., Frey, B., Abowd, G., & Romero, M. (2012). *An evaluation of brailletouch: Mobile touchscreen text entry for the visually impaired* [Paper presentation]. Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services (pp. 317–326).

Sun, K., Yu, C., & Shi, Y. (2019). Exploring low-occlusion qwerty soft keyboard using spatial landmarks. *ACM Transactions on Computer-Human Interaction*, *26*(4), 1–33. https://doi.org/10.1145/3318141

Vertanen, K. (2009). *Efficient correction interfaces for speech recognition* [Unpublished doctoral dissertation]. Citeseer.

Wang, Y., Yu, C., Liu, J., & Shi, Y. (2013). *Understanding performance of eyes-free, absolute position control on touchable mobile phones* [Paper presentation]. Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services (pp. 79–88). https://doi.org/10.1145/2493190.2493215

Xu, W., Liang, H.-N., Zhao, Y., Zhang, T., Yu, D., Monteiro, D., & Yue, Y. (2019). Ringtext: Dwell-free and hands-free text entry for mobile head-mounted displays using head motions. *IEEE Transactions on Visualization and Computer Graphics*, *25*(5), 1991–2001. https://doi.org/10.1109/TVCG.2019.2898736

Ye, L., Sandnes, F. E., & MacKenzie, I. S. (2020). *Qb-gest: Qwerty bimanual gestural input for eyes-free smartphone text input* [Paper presentation]. International Conference on Human-Computer Interaction (pp. 223–242).

Yu, D., Fan, K., Zhang, H., Monteiro, D., Xu, W., & Liang, H.-N. (2018). Pizzatext: Text entry for virtual reality systems using dual thumbsticks. *IEEE Transactions on Visualization and Computer Graphics*, *24*(11), 2927–2935. https://doi.org/10.1109/TVCG.2018.2868581

Zhao, M., Pierce, A. M., Tan, R., Zhang, T., Wang, T., Jonker, T. R., … Gupta, A. (2023). *Gaze speedup: Eye gaze assisted gesture typing in virtual reality* [Paper presentation]. Proceedings of the 28th International Conference on Intelligent User Interfaces (pp. 595–606). https://doi.org/10.1145/3581641.3584072

Zhu, S., Zheng, J., Zhai, S., & Bi, X. (2019). *i'sfree: Eyes-free gesture typing via a touch-enabled remote control* [Paper presentation]. Proceedings of the 2019 Chi Conference on Human Factors in Computing Systems (pp. 1–12).

Zhu, Y., Tian, D., & Yan, F. (2020). Effectiveness of entropy weight method in decision-making. *Mathematical Problems in Engineering*, *2020*, 1–5. https://doi.org/10.1155/2020/3564835

About the authors

Aoxin Sun is a master student in the School of Computer Science and Engineering of Beihang University, China. His current research focuses on virtual reality, augmented reality, and HCI.

Lili Wang received her PhD degree from Beihang University, Beijing, China. She is a professor with the School of Computer Science and Engineering of Beihang University and a researcher with the State Key Laboratory of Virtual Reality Technology and Systems. Her interests include virtual reality, real-time rendering and HCI.

JiaYe Leng is a PhD student in the School of Creative Media, City University of HongKong, China. His current research focuses on virtual reality, augmented reality, and HCI.

Sio Kei Im received his PhD degree from Queen Mary University of London, United Kingdom. He is a professor at the Faculty of Applied Sciences, Macau Polytechnic University. His research interests include video coding, image processing and machine learning.