GazeFlick: A Practical Text Input Method for AR-HMD Combining Eye Gaze and Flick Gesture*

Takumi Kishi[1], Yuichiro Fujimoto[1,*]

[1]*Ryukoku University, 1-5, Yokoya, Ooe-Cho, Seta, Otsu, Shiga, 520-2123, Japan*

Abstract

Although research has been conducted for many years on text input while wearing a head-mounted display (HMD) for augmented reality (AR), a de facto standard has not yet been established. Many HMDs recently have begun to adopt GazePinch, a method for selecting distant objects by pinching with a finger while the user's gaze is fixed on the object. On the other hand, "flick input" is mainly used for inputting Japanese and other languages on smartphones, and it is known to be a fast text input method once the user gets used to it. In this study, we propose GazeFlick, a combination of GazePinch and flick input, as a new text input method for AR/VR-HMD. Preliminary experiments have confirmed a significant improvement in input speed compared to a hand ray method with a common virtual keyboard.

Keywords

Augmented reality, virtual reality, text input, eye gaze

1. Introduction

Text input while wearing a head-mounted display (HMD) for augmented reality (AR) and virtual reality (VR) has been a longstanding issue in the field. When using AR/VR-HMDs while standing, traditional keyboards are difficult to use, so alternatives are needed. As of 2024, many AR/VR-HMDs have virtual keyboard functionality (a common QWERTY layout) as a standard text input feature. Input is performed by performing a decision operation (i.e., pressing a button on the controller or tapping a finger) with the ray from the VR controller or actual hand against each key. For the majority of users accustomed to the general QWERTY keyboard layout, there is little major confusion, but there is a problem that the operation does not become much faster even if the user gets used to it. Many researchers and developers have made numerous proposals over the past decade in search of better operating methods (e.g., [1, 2, 3, 4]). However, as of 2024, there is still no text input method that can become the de facto standard.

The GazePinch [5], which determines the selection of distant objects by pinch movements of the finger while the user's gaze is directed toward the object, has begun to be used in many devices (e.g., Microsoft HoloLens2 with MRTK3, Apple Vision Pro, etc.). On the other hand, "flick input" is mainly used to input Japanese and other languages on touchpad devices such as smartphones. Flick input focuses on the fact that each Japanese character (hiragana and katakana) can be grouped into five chunks, and only the primary keys for each group are displayed on the main screen. When the user touches each primary key, the other keys in the group are displayed around it and the key at the position where the user takes his/her finger off the screen is input. In other words, to input the surrounding keys, the user must translate his/her finger either up, down, left, or right, which is called a "flick" action. This is known to be a fast text input method once the user gets used to it. This method can also be easily extended to other languages input, as long as some keys can be grouped.

In this research, we focus on GazePinch and flick input, and propose a new input method for AR-HMD that combines them. Specifically, the appearance of the UI follows that of the UI for flick input, and only the main keys of each group are displayed. When the user looks at one of them and performs a pinch motion (same as GazePinch), the other keys of that group are displayed above, below, left, or right of that main key. With the user's thumb and index finger closed, move two fingers in any of the four directions (up, down, left, or right), and the key that was selected when the fingers were separated will be input. We implemented the proposed method on HoloLens2 and conducted preliminary experiments comparing it with the general method of using a hand ray and QWERTY virtual keyboard, and obtained promising results showing a significant improvement in input speed.

2. Related Work

There have been many methods proposed for text input on AR/VR-HMDs. In this section, we will discuss the methods that are particularly relevant to this research,

APMAR'24: The 16th Asia-Pacific Workshop on Mixed and Augmented Reality, Nov. 29-30, 2024, Kyoto, Japan

*Corresponding author.

y210131@mail.ryukoku.ac.jp (T. Kishi); yfujimoto@rins.ryukoku.ac.jp (Y. Fujimoto)
https://yfujimoto.cfbx.jp/ (Y. Fujimoto)
0000-0002-8270-2609 (Y. Fujimoto)

focusing on recent years (from 2020 onwards).

Streli et al. proposed TapType, which allows any flat surface to be used as a pseudo-keyboard [6]. In this method, the user's tapping motion on the flat surface is decoded by an inertial sensor built into a wristband, and is associated with the layout of the conventional QWERTY keyboard. Lu et al. devised a method for entering text without using the hands using an AR-HMD with three types of gaze input (dwell, trajectory, and blinking). Interestingly, this method does not display a virtual keyboard, but instead has the user operate it while imagining a standard QWERTY keyboard. By combining these, it overcomes the problem of arm fatigue during input and the problem of the virtual keyboard blocking the real environment [7].

Several methods have been proposed to reduce the effort required for direct input by introducing a probability model. Fashimpaur et al. proposed the PinchType method, which uses a standard QWERTY keyboard and pinch gestures [4]. All the keys are grouped into eight types, corresponding to the eight fingers of the hand, including the thumb. Since there are more than eight letters in the alphabet, there is some ambiguity, but this method dynamically resolves this ambiguity during input by introducing a language model. Adhikary and Vertanen proposed a text input method that enables fast and error-free text input by adding a function for automatically suggesting correction candidates using text context to a general virtual keyboard, and supporting these functions using speech recognition [8]. Dudley et al. proposed a keyboard that uses a probability model that allows for the uncertainty of the imperfect hand tracking function of consumer VR-HMDs. Using this, they compared the peak performance of the general input method of poking with the index finger and the long-distance input of ray casting, and analyzed them in detail [9].

Takahashi et al. have also introduced flicking as a method of inputting Japanese for a VR-HMD [10]. They use a VR controller with a trackpad for flick input, instead of the GazePinch idea, which is different from us. By using a trackpad, the operation is closer to flick input on a smartphone, so when used by people with experience in flick input, it was shown that text can be input faster than with a QWERTY keyboard using a ray. In June 2024, Apple Vision Pro was launched in Japan, and a method similar to the proposed method was adopted for inputting Japanese. However, due to its special camera arrangement, Apple Vision Pro is capable of hand tracking over a much wider range and at a much faster speed than HoloLens2 and other HMDs. Furthermore, it is known to have a fairly high level of accuracy in eye tracking. This research is mainly based on the assumption that HoloLens2 will be used as the device, and it will be tuned as a text input method suitable for limited hand tracking and eye tracking performance.

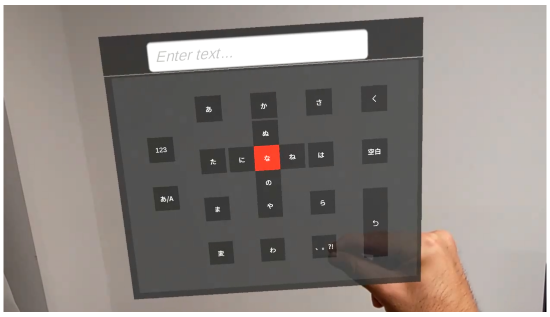

Figure 1: Overview of GazeFlick. The user is looking at it, and the key that has been pinched is changing to red. The sub-keys are expanding in the upper, lower, left and right directions.

3. Method

3.1. Idea

The proposed method GazeFlick combines GazePinch [5] and flick input (Figure 1). One factor that contributes to the complexity of text entry is the number of inputs. Especially in Japanese, it takes, on average, two Roman alphabet characters to input one character of the smallest unit (i.e., hiragana) in the most common Roman input method. While this is not a major problem for physical keyboards that allow high-speed input, it is fatal for virtual keyboards. On the other hand, there is a direct input method that displays all 45 types of characters in all the smallest units and allows the user to input them once, but few Japanese prefer this method because it takes time to find a large number of characters. On the other hand, in smartphones, which contain similar problems, "flick input" solves these problems at the same time.

It is assumed that the HMD worn by the user is a device that can perform eye-gaze measurement and hand tracking. Each key is grouped by no more than five and has an ordering relationship among them. Only the youngest key in the group is displayed on the screen (this is called the primary key). First, as with GazePinch, the collision between the gaze and the main key is determined by measuring the gaze (Figure 2). When Gaze collides with a specific primary key and a pinch operation (index finger and thumb in contact) is performed, other keys (called subkeys) in the group are displayed in order in the four locations to the left, above, right, and below the primary key. With the index finger and thumb in contact, moving the hand up, down, left or right from that position highlights the corresponding subkey. When the index finger and thumb are released, the highlighted key is entered and all sub keys disappear once. Alternatively, if the user pinches the main key and then separates their thumb and forefinger without sliding their hand, the character on the main key will be entered. The user repeats this

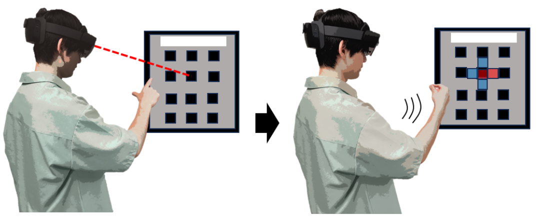

Figure 2: How to use the GazeFlick. The red ray of the gaze is not actually presented to the user.

process to input the data.

3.2. Implementation

Microsoft HoloLens2 was used for this implementation in this paper. However, the proposed method can be easily applied to other devices that are capable of eye measurement and hand tracking. Unity2022.2.18f and MRTK3 (3.2.0) were used as the software for the implementation.

As a use case, we have implemented hiragana input for Japanese, but it can easily be replaced with other languages as well. Hiragana is based on units of 46 characters, and is divided into 11 groups containing 5 or fewer characters (Figure 3). In addition, there are characters with voiced consonants and semi-voiced consonants derived from them. After selecting each character using pinch operation, the user can input the corresponding voiced consonants by pushing his/her finger forward, and he/she can input the corresponding semi-voiced consonants by pulling his/her finger back.

Pinching was based on the vertical plane of the HMD position posture at the time of tapping, and was recognized as an up/down or left/right movement when the finger was moved beyond a certain distance (currently 50 mm) from it, respectively. The key that the user is looking at is set to change to a color with higher brightness. During the pinch, the transparency of the other keys and the back panel was increased to 10 % to make the key that is being focused on stand out. After pinching, the currently selected key changes to red to make it more noticeable. The window in which the characters are displayed is fixed at the world coordinate system position that called it, and only follows when the user moves a lot (i.e., Billboard display provided by MRTK). The detailed parameter settings for implementation are as shown in Table 1.

4. Pre-User Study

In order to check the efficiency of text input for the proposed method, a pre-user study was conducted prior to the formal user study scheduled for the near future.

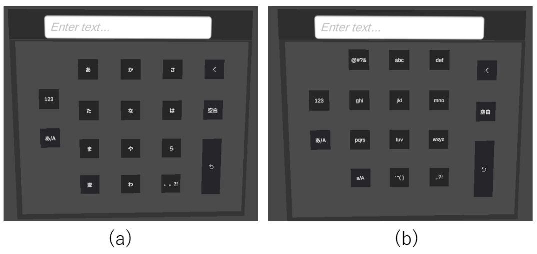

Figure 3: Keyboard layout. (a) Japanese hiragana keys, (b) English alphabet keys. Both modes can be switched using the button in the lower left corner.

Table 1
Parameters of the current implementation

Parameters	Value
Distance from user	1.5 m
Size of each button	75 mm × 75 mm
Distance between each button	90 mm
Whole panel size	0.9 m × 0.7 m
Gaze dwell for button active	0.05 s
Arm movement for flick	50 mm

Please note that only one researcher participated in the pre-user study. He is a native speaker of Japanese, used a smartphone on a daily basis and used flick input to enter text. In addition, he was also using HoloLens2 on a daily basis and was proficient in hand-ray operation and the proposed method (GazeFlick).

We positioned the proposed method as one for use over long distances, and compared it with the conventional method of inputting using hand ray and two-finger tapping. For the experiment, two Japanese character sets were created: (a) hiragana set (no voiced or semi-voiced consonants), (b) hiragana set (with voiced and semi-voiced consonants). As an example, one of the strings in (b) is “Bo ku no pa pa ha ma n ga ka de su (My dad is a manga artist)". As mentioned above, for the input of voiced and semi-voiced sounds, the proposed method requires the pinch to be moved in the depth direction, so we created different character sets for these sounds. The input characters are displayed at the top of the keyboard, and the input begins. The time it took to type the 12 characters was measured. If a character was input incorrectly, it was possible to delete it using the backspace key. A total of five attempts were made for each string. The two input methods were used in random order for each string.

The results of the preliminary experiment are shown below as the average of 3 different strings x 5 times (number of trials) = 15 trials in one set. For (a) hiragana set (no voiced or semi-voiced consonants), Hand-ray (conventional method) took 21.8 s, and GazeFlick (proposed method) took 15.2 s. For (b) hiragana set (with voiced or semi-voiced consonants), no alphanumeric characters), Hand-ray took 21.2 s, and GazeFlick (proposed method) took 16.9 s.

The results showed that when used by experienced users, the proposed method has the potential to enable faster character input than hand-ray. In particular, when entering characters that include voiced and semi-voiced consonants, it was expected that the input speed with GazeFlick would be somewhat slower because the additional pinch gesture would need to be moved in the depth direction, but the speed was almost same. In addition to Japanese, the input of additional elements for certain basic characters can be seen in multiple other languages (Thai, Arabic, French, etc.). The fact that the proposed method showed that it was possible to input voiced and semi-voiced consonants smoothly without significantly reducing the movement speed is expected to be beneficial in the use of these multiple languages as well.

5. Summary and Future Work

In this study, we proposed GazeFlick as a method of entering text that is an extension of GazePinch, which uses eye movement and flicking. Preliminary experiments (with only one person) showed that, with practice, this method could be 20 % to 30 % faster than the conventional hand-ray method. In the future, we plan to conduct formal user studies and clarify the applicability of the system to multiple languages (particularly English).

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Number JP22K12127.

References

[1] J. J. Dudley, K. Vertanen, P. O. Kristensson, Fast and precise touchbased text entry for head-mounted augmented reality with variable occlusion, ACM Transactions on Computer-Human Interaction 25 (2018).

[2] S. Ahn, S. Heo, G. Lee, Typing on a smartwatch for smart glasses, in: Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces, ISS '17, ACM, 2017, pp. 201–209.

[3] W. Xu, H.-N. Liang, Y. Zhao, T. Zhang, D. Yu, D. Monteir, Ringtext: Dwell-free and hands-free text entry for mobile head-mounted displays using head motions, IEEE Transactions on Visualization and Computer Graphics 25 (2019) 1991–2001.

[4] J. Fashimpaur, K. Kin, M. Longest, Pinchtype: Text entry for virtual and augmented reality using comfortable thumb to fingertip pinches, in: Extended Abstracts of the 2020 Conference on Human Factors in Computing Systems, CHI EA '20, ACM, 2020, pp. 1–7.

[5] K. Pfeuffer, B. Mayer, D. Mardanbegi, H. Gellersen, Gaze + pinch interaction in virtual reality, in: Proceedings of the 5th Symposium on Spatial User Interaction, SUI '17, ACM, 2017, pp. 99–107.

[6] P. Streli, J. Jiang, A. Fender, M. Meier, H. Romat, C. Holz, Taptype: Ten-finger text entry on everyday surfaces via bayesian inference, in: Proceedings of the 2022 Conference on Human Factors in Computing Systems, CHI '22, ACM, 2022.

[7] X. Lu, D. Yu, H.-N. Liang, J. Goncalves, itext: Hands-free text entry on an imaginary keyboard for augmented reality systems, in: The 34th Annual ACM Symposium on User Interface Software and Technology, UIST '21, ACM, 2021.

[8] J. Adhikary, K. Vertanen, Text entry in virtual environments using speech and a midair keyboard, IEEE Transactions on Visualization and Computer Graphics 27 (2021) 2648–2658.

[9] J. J. Dudley, J. Zheng, A. Gupta, H. Benko, M. Longest, R. Wang, P. O. Kristensson, Evaluating the performance of hand-based probabilistic text input methods on a mid-air virtual qwerty keyboard, IEEE Transactions on Visualization and Computer Graphics 29 (2023) 4567–4577.

[10] R. Takahashi, S. Shirai, J. Orlosky, Y. Uranishi, H. Takemura, A japanese character flick-input interface for entering text in vr, in: 2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), ISMAR2021, IEEE, 2021, pp. 261–263.