FistPointer: Target Selection Technique using Mid-air Interaction for Mobile VR Environment

Akira Ishii
Takuya Adachi
Keigo Shima
Shuta Nakamae
Buntarou Shizuki
Shin Takahashi
University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki
305-8573, Japan
{ishii@iplab.,tadachi@acs.,keigo@iplab.,
nakamae@iplab.,shizuki@,shin@}cs.tsukuba.ac.jp

Abstract
We present FistPointer, a target selection technique using mid-air interaction behind a smartphone for mobile virtual reality (VR) environment, which is realized by adopting a cardboard viewer with a smartphone as a head-mounted display. Our technique displays a pointer on the screen corresponding to the position of the hand detected by the built-in back camera of the smartphone. The user can move the pointer by moving the hand in a thumbs-up posture and select a target by folding the thumb similar to pushing the button of a joystick. Our technique can be implemented using only the built-in camera of a smartphone, thus it is easy to apply our technique to a target selection in mobile VR environments. To evaluate the performance of our technique, we conducted an experiment in selection task. Furthermore, we developed a game using our technique and investigated the user impressions.

Author Keywords
Virtual reality; head-mounted display; cardboard; pointing; targeting.

ACM Classification Keywords
H.5.2 [Information Interfaces and Presentation (e.g. HCI)]: User Interfaces – Input devices and strategies.

Figure 1: Overview of FistPointer.

Figure 2: Hand postures under FistPointer. Upper: Pointing. Lower: Selection (i.e., click). The hand posture of click is a metaphor of pushing a button.

Introduction

Mobile virtual reality (VR) environment using a cardboard viewer and smartphone as a head-mounted display (HMD), such as Google Cardboard [2] and HACOSCO [6], opens the door to new possibilities where people can experience VR easily and inexpensively. However, such mobile VR environment covers the touch screen of the smartphone and thus restricts the user from touching the screen. Therefore, it is difficult to realize applications which need intensive interaction such as games.

To address this problem, we present FistPointer (Figure 1), a target selection technique using mid-air interaction behind the mobile device. Our technique displays a pointer on the screen corresponding to the position of the hand detected by the built-in back camera of the mobile device. The user can move the pointer by moving the hand in a thumbs-up posture, as shown in Figure 2 (upper). The user also can select a target by folding the thumb similar to pushing the button of a joystick, as shown in Figure 2 (lower). The hand posture of click is a metaphor of pushing a button. Our technique can be implemented using only a smartphone with a built-in back camera, thus it is easy to apply our technique to target selection in mobile VR environments.

In this study, we prototyped a proof-of-concept implementation of FistPointer with common technologies and conducted a pilot study with three target size conditions to investigate the target size that a user can select. Furthermore, we developed a game using our technique and investigated the user impression.

Related Work

Target Selection for Large Displays

Numerous studies that investigated target selection techniques for large displays have been explored [4]. For example, Vogel and Balakrishnan [17] proposed three techniques for gestural pointing and two for clicking that are designed for large displays. Markussen et al. [12] proposed three text entry techniques using mid-air interaction for large displays. By contrast, we focus on designing the selection technique for mobile VR environments.

Target Selection using Camera of Mobile Device

Baldauf et al. [1] proposed a target selection technique using the position of the detected fingertip. While their study would be the most closely related to ours, ‘click’ was not described. By contrast, we design both pointing and clicking using different hand postures. GUI operation techniques using two fingers [11] and using hand gestures in mid-air [15] were proposed. They used hand gestures as triggers for specific actions. By contrast, our technique uses the hand for target selection. Numerous studies that achieved pointing by attaching an external camera to a mobile device also have been explored [13, 18]. In our technique, we achieve target selection only by using the built-in back camera of the mobile device, thus it does not require any external device.back

Target Selection for VR

Numerous techniques for VR have been investigated to enable selection of targets displayed on an HMD. Sugiura et al. [16] proposed a technique to select a virtual button displayed on the HMD by a fingertip. Kato et al. [7] proposed a target selection technique by a hand gesture for HMDs, and evaluated the selection speed compared to a wireless mouse. Lee et al. [9] and Gugenheimer et al. [5] proposed virtual space interaction technique by detecting the user’s touch operation on a touch screen installed in front of the HMD. Petry et al. [14] proposed an operation technique of panorama video by using the horizontal movements of the hand detected by a Leap Motion sensor installed in the front

side of the HMD. These previous studies require external devices. By contrast, our technique uses only the built-in camera of the mobile device to detect a hand. Therefore, it is easy to apply our technique to mobile VR environments.

FistPointer

FistPointer is a target selection technique where the user can point at a target by moving the hand in the posture as shown in Figure 2 (upper). The user can select the target by folding the thumb as shown in Figure 2 (lower), which is the gesture designed using a metaphor of pushing a button and adopted by Gunslinger [10] and investigated in menu selection task [8].

Hand Detection

First, our technique determines whether each pixel of the RGB image belongs to the hand or not. To implement this, we use the algorithm of Song et al. [15]. As the result, we obtain a binary image with the hand shape as shown in Figure 3.

Calculation of Pointer Coordinates

We display the pointer at the base of the index finger. This design prevents the pointer from wobbling when clicking.

To determine the pointer coordinates (x_p, y_p), first y_p is calculated, and then x_p is calculated from y_p. Firstly, our technique evaluates the width of the hand for each y coordinate. Then, in the i^{th} y coordinate, the slope of the regression line (Figure 3 (left)) of the hand width is calculated from the past 9 y coordinates $(y_i, ..., y_{i-8})$ including i. This is calculated for every y below the tip of the thumb. The wave in Figure 3 (middle) shows the magnitude of slopes. In this figure, the slopes near the tip of the thumb and the base of the thumb are steep. Moreover, the slope of the base is steeper than the tip. Therefore, we use the y coordinate of the peek of the wave as y_p. Then, x_p (the position of the red dot shown in Figure 3 (middle)) is defined as the right edge of the hand in y_p.

Note that our technique requires the whole hand to be captured by a camera because our technique uses the width of the hand to determine the pointer coordinates and the tip of thumb to detect clicks. As the result, the range of the possible pointer coordinates is smaller than that of the mapped camera image, as illustrated as the yellow rectangle in Figure 3 (right). Due to this, we map (x_p, y_p) to the mapped pointer coordinates (x'_p, y'_p) using Equations (1) and (2) (when the display size is 1920 $\times$ 1080 pixels).

$$x'_p = \begin{cases} 0 & (x_p < 70) \\ (x_p - 70) \times 8 & (70 \leq x_p \leq 310) \\ 1920 & (x_p > 310), \end{cases} \quad (1)$$

$$y'_p = \begin{cases} 0 & (y_p < 55) \\ (y_p - 55) \times 8 & (55 \leq y_p \leq 190) \\ 1080 & (y_p > 190). \end{cases} \quad (2)$$

Detection of Click

Our technique recognizes that a click is performed when the tip of the thumb comes near the pointer coordinates. To detect this, the system finds the coordinates of the tip of the thumb (x_t, y_t): the system defines the smallest y coordinate of the hand as y_t, and then the middle point of the hand of y_t as x_t. Our technique calculates the value that is the width of the hand at y_p divided by the length of the thumb (the distance between y_t and y_p). If this value exceeds the threshold (the threshold is 1.6, which we determined experimentally in this study), our technique recognizes the user operation as click. This design can deal with variation in the size of the hand captured by the camera.

Figure 4: Hacosco and smartphone.

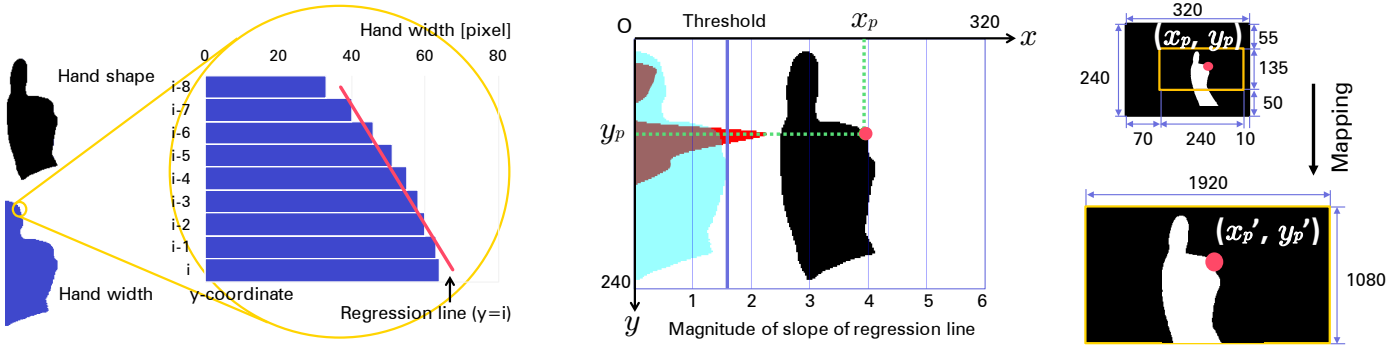

Figure 3: Left: Regression line (red line) at y-coordinate i. Middle: Calculation of pointer coordinates. Hand shape (black). The width of the hand (blue). The magnitude of slope of regression line (red). The red dot is the position of pointer. Right: Mapping between an image captured by a camera and pointer coordinate.

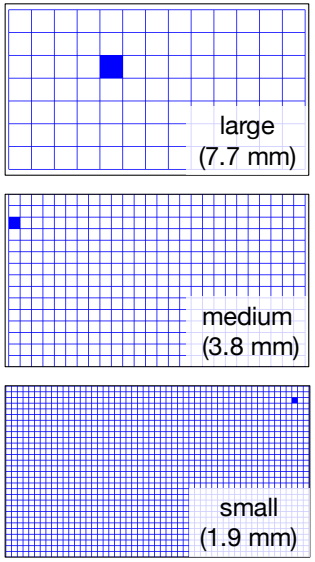

Figure 5: Target size conditions.

Prototype

We implemented a prototype of FistPointer on an Android smartphone (SONY Xperia Z5), as shown in Figure 4. In this implementation, the camera image was captured at 14.3 fps. The pointer moves at 60.2 fps with the latency of 177 ms; the latency of clicking was 170 ms. These latency values were observed using a high-speed camera (SONY DSC-RX100M5; 960 fps in 1920 $\times$ 1080 pixels).

Pilot Study

We conducted a pilot study to investigate the target size that a user can select accurately using FistPointer in a mobile VR environment. We made targets of three different sizes and measured the target selection speed and accuracy.

Participants and Apparatus

Eight volunteers (all males, our laboratory members) aged between 22 and 24 years (mean: 23.1) participated in the pilot study. All participants had corrected vision; 7 wore glasses and 1 wore contact lenses.

We used an Android smartphone (Xperia Z5, dimensions: 146 mm $\times$ 72 mm $\times$ 7.3 mm, display size: 5.2 inch, resolution of screen: 1920 $\times$ 1080 pixels, OS: Android 6.0.1) for the pilot study. We also used a monocular lens type VR goggle called HACOSCO Tatami 1 [6] as an HMD; we used a monocular one in order to reduce possible effects caused by the binocular vision (e.g., VR sickness, adjustment of parallax).

Experimental Design

To evaluate the effect by the target size to the target selection speed and accuracy, we designed targets of three different sizes (*target size conditions*) shown in Figure 5.

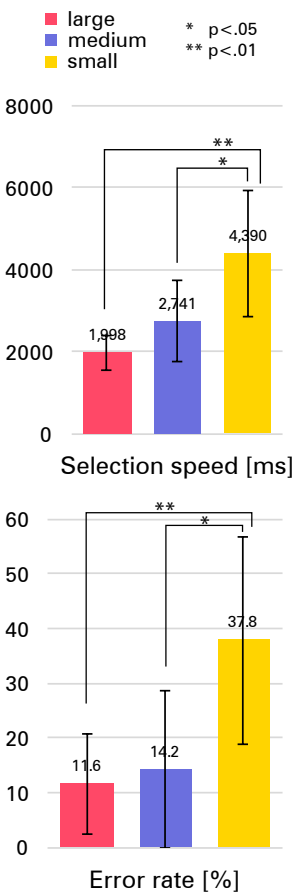

Figure 6: Result of the pi lot study. Error bars indicate ± one SD.

A target is the blue colored square cell in the grid. We defined the largest target size as 48 dp, which is the smallest target size that Google recommends in the guidelines [3], and prepared three target sizes, 48 dp, 24 dp (the half of 48), 12 dp (the half of 24), which are 7.7, 3.8, and 1.9 mm, respectively. Hereinafter, we call these the large, medium, and small conditions. As shown in Figure 5, the number of column and row of the grid of the three conditions are 7×13, 15×26, and 30×53.

Procedure and Task

We conducted the experiment in a calm office environment that was well-lighted by fluorescent lights. The background was clean, colored in white, with no pattern, and with no shadow. In order to control the experimental condition, the experimenter asked the participant to hold the HMD with the left hand and to use our technique with the right hand. Then the participant repeated selecting the center and four corners in the large condition as a training session until the participant became familiar with our technique.

The participant selected a target as a trial. The participant was instructed to select the target as quickly and accurately as possible. The next target was displayed when the participant succeeded in the current target selection. To eliminate the order effect, the targets were presented in a random order without redundancy. The participant selected 20 targets under each size condition as one session. The session started when the participant succeeded in the first dummy target selection. The participant performed each session once for the large, medium, and small conditions in order. Therefore, the participant performed 60 trials ($= 20$ trials $\times$ 3 size conditions) in total. To reduce the effect of fatigue, the participant took a break longer than a minute when each session was finished.

After all the sessions were finished, the experimenter asked the participant to fill out a questionnaire to investigate the impression. This experiment took approximately 20 minutes per participant, including the prior briefing and responding to the questionnaire.

Results

The selection speed and error rate for each target size condition are shown in Figure 6. We analyzed the results with a repeated measure ANOVA and Tukey multiple comparisons at 5% significance level. The within-subject factor is the target size.

Selection Speed

We observed a significant main effect for the selection speed ($F_{2,\ 21} = 10.21$, $\eta^2 = 0.49$, $p < 0.001$). Post-hoc tests revealed the followings: 1) a significant difference was found between the large and small conditions ($p < 0.001$); 2) a significant difference was found between the medium and small conditions ($p < 0.05$); 3) there was no significant difference between the large and medium conditions; and 4) the small condition showed a significantly slowest selection speed among all size conditions.

Error Rate

We calculated the error rate which is the number of errors divided by the total number of selections. We observed a significant main effect within the error rate ($F_{2,\ 21} = 7.71$, $\eta^2 = 0.42$, $p < 0.01$). Post-hoc tests revealed the followings: 1) a significant difference was found between the large and small conditions ($p < 0.01$); 2) a significant difference was found between the medium and small conditions ($p < 0.05$); 3) there was no significant difference between large and medium; and 4) the small size condition showed a significantly highest error rate among all size conditions.

Discussion
From the participants' comments, the error rate might have bias between target regions. One participant commented "*The targets in the left side of the screen were more difficult to select than the ones in the right side*"; two participants commented "*It was difficult to select the edge and corner of the screen.*" The reason would be that the participants had to stretch their arm to the limits because the left side of the screen was the farthest from the right.

Test with an Application: Shooter Game

We developed a shooter game using FistPointer as shown in Figure 7. Players can move the target scope by moving their hand and can shoot by folding their thumb ('click').

Figure 7: Shooter game using FistPointer in mobile VR environment.

Nine volunteers (1 female, including 3 of our laboratory members) aged between 21 and 25 years (mean: 23.6) played this game as participates. After finishing the game, we asked the participants to answer impressions. We obtained various feedback. Three participants commented that "*I was able to select a target easily because the accuracy of FistPointer was high*"; 2 participants commented that "*FistPointer was easy to understand because the operation of folding my thumb (click) was related to the operation of shooting a gun.*" By contrast, 3 participants commented that "*I thought that my arm will get tired if I play for a long time*" and one of them commented that "*I could operate more accurately when I extend my arm. However, I think that I will get tired faster by extending my arm. This problem might be solved by using a camera with wider field of view.*"

The above comments also support our observations: the hand posture of our technique is easy to recall. This is because our technique employs stretching users' arm in front of their eyes to select a target and folding their thumb to click it, which a metaphor of pushing a button.

Conclusions

We present a target selection technique using mid-air interaction behind a smartphone for mobile VR environments. Our technique can be implemented using only the built-in camera of a smartphone, thus it is easy to apply our technique to a target selection in mobile VR environments. To evaluate the performance of our technique, we conducted a pilot study in selection task and explored selection speed and accuracy of our technique under the three targets of different size conditions. Based on the result of the pilot study, the large (7.7 mm) and medium (3.8 mm) conditions can be used practically. Furthermore, we also developed a game using our technique and investigated the user impressions.

Future Impact

By using FistPointer and the cardboard viewer, users will be able to enjoy active VR contents anywhere easily. For VR creators, FistPointer allows them to create more active and thus flexible contents. We believe this study expands the possibilities of mobile VR.

References

[1] Matthias Baldauf, Sebastian Zambanini, Peter Fröhlich, and Peter Reichl. 2011. Markerless Visual Fingertip Detection for Natural Mobile Device Interaction. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. ACM, New York, NY, USA, 539–544. DOI: http://dx.doi.org/10.1145/2037373.2037457

[2] Google. 2015. Google VR Cardboard. (2015). https://vr.google.com/cardboard/ (accessed on 11 January 2017).

[3] Google. 2016. Google design guidelines: Button.

(2016). https://material.google.com/components/buttons.html (accessed on 11 January 2017).

[4] Celeste Groenewald, Craig Anslow, Junayed Islam, Chris Rooney, Peter Passmore, and William Wong. 2016. Understanding 3D Mid-Air Hand Gestures with Interactive Surfaces and Displays: A Systematic Literature Review. In *Proceedings of the British Human Computer Interaction Conference (British HCI '16)*. 13 pages.

[5] Jan Gugenheimer, David Dobbelstein, Christian Winkler, Gabriel Haas, and Enrico Rukzio. 2016. FaceTouch: Enabling Touch Interaction in Display Fixed UIs for Mobile Virtual Reality. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 49–60. DOI: http://dx.doi.org/10.1145/2984511.2984576

[6] HACOSCO INC. 2015. Smart phone VR Hacosco | Simple VR experience. (2015). http://hacosco.com/en/ (accessed on 11 January 2017).

[7] Haruhisa Kato and Hiromasa Yanagihara. 2013. PACMAN UI: Vision-based Finger Detection for Positioning and Clicking Manipulations. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services (MobileHCI '13)*. ACM, New York, NY, USA, 464–467. DOI: http://dx.doi.org/10.1145/2493190.2494652

[8] Arun Kulshreshth and Joseph J. LaViola, Jr. 2014. Exploring the Usefulness of Finger-based 3D Gesture Menu Selection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1093–1102. DOI: http://dx.doi.org/10.1145/2556288.2557122

[9] Jihyun Lee, Byungmoon Kim, Bongwon Suh, and Eunyee Koh. 2016. Exploring the Front Touch Interface for Virtual Reality Headsets. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '16)*. ACM, New York, NY, USA, 2585–2591. DOI: http://dx.doi.org/10.1145/2851581.2892344

[10] Mingyu Liu, Mathieu Nancel, and Daniel Vogel. 2015. Gunslinger: Subtle Arms-down Mid-air Interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 63–71. DOI: http://dx.doi.org/10.1145/2807442.2807489

[11] Zhihan Lv, Alaa Halawani, Muhammad Sikandar Lal Khan, Shafiq Ur Réhman, and Haibo Li. 2013. Finger in Air: Touch-less Interaction on Smartphone. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia (MUM '13)*. ACM, New York, NY, USA, Article 16, 4 pages. DOI: http://dx.doi.org/10.1145/2541831.2541833

[12] Anders Markussen, Mikkel R. Jakobsen, and Kasper Hornbæk. 2013. *Selection-Based Mid-Air Text Entry on Large Displays*. Springer Berlin Heidelberg, Berlin, Heidelberg, 401–418. DOI: http://dx.doi.org/10.1007/978-3-642-40483-2_28

[13] Takehiro Niikura, Yoshihiro Watanabe, Takashi Komuro, and Masatoshi Ishikawa. 2014. In-Air finger motion interface for mobile devices with vibration feedback. *IEEJ Transactions on Electrical and Electronic Engineering* 9, 4 (2014), 375–383. DOI: http://dx.doi.org/10.1002/tee.21982

[14] Benjamin Petry and Jochen Huber. 2015. Towards Effective Interaction with Omnidirectional Videos Using Immersive Virtual Reality Headsets. In *Proceedings of the 6th Augmented Human International Conference (AH '15)*. ACM, New York, NY, USA, 217–218. DOI: http://dx.doi.org/10.1145/2735711.2735785

[15] Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. 2014. In-air Gestures Around Unmodified Mobile Devices. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 319–329. DOI: http://dx.doi.org/10.1145/2642918.2647373

[16] Atsushi Sugiura, Masahiro Toyoura, and Xiaoyang Mao. 2014. A Natural Click Interface for AR Systems with a Single Camera. In *Proceedings of Graphics Interface 2014 (GI '14)*. Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 67–75. http://dl.acm.org/citation.cfm?id=2619648.2619660

[17] Daniel Vogel and Ravin Balakrishnan. 2005. Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST '05)*. ACM, New York, NY, USA, 33–42. DOI: http://dx.doi.org/10.1145/1095034.1095041

[18] Daniel Wigdor, Clifton Forlines, Patrick Baudisch, John Barnwell, and Chia Shen. 2007. Lucid Touch: A See-through Mobile Device. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*. ACM, New York, NY, USA, 269–278. DOI: http://dx.doi.org/10.1145/1294211.1294259

Akira Ishii
He is currently a graduate student of the Department of Computer Science, Graduate school of Systems and Information Engineering, University of Tsukuba. He graduated from the College of Media Arts, Science and Technology, School of Informatics, University of Tsukuba, Japan. He is interested in the field of human computer interaction.

Takuya Adachi
He is currently a graduate student of the Department of Computer Science, Graduate school of Systems and Information Engineering, University of Tsukuba. He graduated from the College of Media Arts, Science and Technology, School of Informatics, University of Tsukuba, Japan. He is interested in the field of robotics.

Keigo Shima
He is currently a graduate student of the Department of Computer Science, Graduate school of Systems and Information Engineering, University of Tsukuba. He graduated from the College of Media Arts, Science and Technology, School of Informatics, University of Tsukuba, Japan. He is interested in the field of human computer interaction.

Shuta Nakamae
He is currently a graduate student of the Department of Computer Science, Graduate school of Systems and Information Engineering, University of Tsukuba. He graduated from Ibaraki National College of Technology, Japan. He is interested in the field of human computer interaction.