Exploring Indirect Touch Gestures for Smartphone Interaction within VR Environments

Yuto Wada
IPLAB
University of Tsukuba
Tsukuba, Ibaraki, Japan
wada@iplab.cs.tsukuba.ac.jp

Myungguen Choi
IPLAB
University of Tsukuba
Tsukuba, Ibaraki, Japan
choi@iplab.cs.tsukuba.ac.jp

Buntarou Shizuki
IPLAB
University of Tsukuba
Tsukuba, Ibaraki, Japan
shizuki@cs.tsukuba.ac.jp

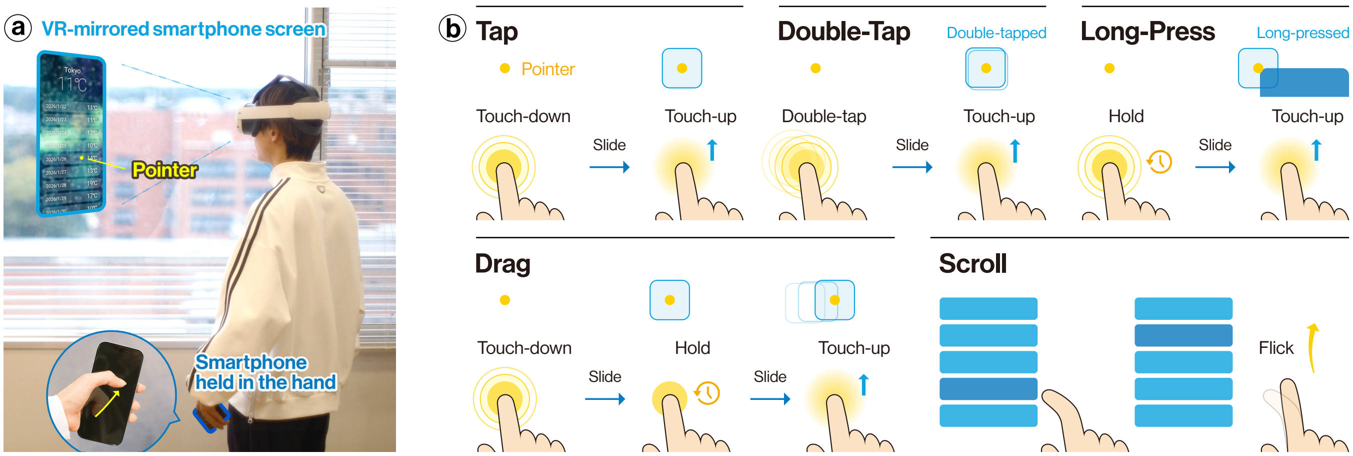

Figure 1: Our indirect touch gestures for smartphone interaction in virtual reality (VR) environments. (a) In indirect touch scenarios, users interact with their smartphones by viewing a mirrored smartphone screen displayed in the VR environment without looking directly at the physical device. (b) Our indirect touch gesture set, which includes a tap gesture adapted for indirect touch, as well as double-tap, long-press, drag, and scroll gestures designed to avoid misrecognition.

Abstract

We present a set of indirect touch gestures that enable smartphone interaction in virtual reality (VR) environments. Previous studies have primarily examined tap gestures in indirect touch scenarios, leaving other essential smartphone gestures (double-tap, long-press, drag, and scroll gestures) largely unexplored. By designing these gestures, we enable users to interact with their smartphones while viewing a mirrored screen in VR environments, without needing to look directly at the physical device, as if they were directly viewing and operating it. We conducted a pilot study to investigate appropriate parameters for these gestures and to develop a model for classifying them. Our results show that the model can classify tap and scroll gestures with an AUC-ROC of 0.997 using a 200 ms window.

CCS Concepts

• **Human-centered computing → Pointing**; **Virtual reality**.

CHI EA '26, Barcelona, Spain

ACM ISBN 979-8-4007-2281-3/26/04
https://doi.org/10.1145/3772363.3798511

Keywords

Head-mounted Display, Touch Screen, Gesture Recognition, Pointing

ACM Reference Format:
Yuto Wada, Myungguen Choi, and Buntarou Shizuki. 2026. Exploring Indirect Touch Gestures for Smartphone Interaction within VR Environments. In *Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems (CHI EA '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3772363.3798511

1 Introduction

It is common for users to want to access information on smartphones when engaging in virtual reality (VR) experiences using head-mounted displays (HMDs) [18, 31]. In such situations, users typically either remove the HMD or rely on video see-through to view their smartphones within the VR environment. However, removing the HMD is cumbersome, while video see-through often suffers from low resolution and visual distortion, both of which limit comfortable interaction. To address these issues, previous studies [6, 30, 31] and commercial systems [5, 14] have proposed mirroring the smartphone display within the VR environment.

The mirrored smartphone screen is typically anchored either to the smartphone itself or within the VR environment. In the former case, users interact with their smartphones while looking directly at the device (*direct touch scenario*). This enables intuitive smartphone operation; however, it requires users to lift the smartphone or look down, which can cause physical fatigue. Moreover, it requires continuously tracking the smartphone's position and orientation, which in turn necessitates additional setup [12, 18]. In the latter case, users interact with their smartphones while viewing the mirrored screen without looking directly at the physical device (*indirect touch scenario* [8, 17, 23], Fig. 1a). In this scenario, users do not need to lift the smartphone or perform any additional setup to track it. However, performing precise tap gestures becomes difficult because the spatial relationship between the screen and the user's finger is unclear [28].

To enable precise interaction in indirect touch scenarios, we present a set of indirect touch gestures for smartphone interaction, which includes tap, double-tap, long-press, drag, and scroll gestures (Fig. 1b). Previous studies have proposed a tap gesture that can be precisely performed in indirect touch scenarios by allowing users to tap the input surface while referencing a pointer displayed on the output surface that indicates the current finger position [1, 13, 15, 19, 27]. However, other gestures that are essential for smartphone interaction [3, 11] have not yet been examined in indirect touch scenarios. By designing other essential smartphone gestures, we aim to enable smartphone interaction in VR environments without requiring users to look at their hands or tracking the smartphone.

To optimize these gestures, we conducted a pilot study to investigate the appropriate parameters for long-press and drag gestures and to develop a classification model for tap and scroll gestures. First, long-press and drag gestures differ from tap gestures in that they require the finger to be held down after touch-down or before touch-up. By collecting tap gesture data, we analyzed the finger-holding duration that can be reliably distinguished from these gestures. Next, both tap and scroll gestures involve a sequence of finger touch-down, movement, and touch-up, which makes them difficult to distinguish. However, because these gestures exhibit distinct characteristics in their velocity profiles [10, 20, 21], we hypothesized that it would be possible to develop a model capable of classifying them. The results showed that a finger-holding duration of 210 ms or longer was optimal for long-press gestures, while a duration of 950 ms or longer was optimal for drag gestures. In addition, the model classified tap and scroll gestures with the area under the receiver operating characteristic curve (AUC-ROC) of 0.997 using data from the first 200 ms after touch-down. The implementation of our gesture-recognition system has been released on GitHub[1].

2 Related Work

In this section, we review related work on indirect touch interaction and interaction techniques for operating smartphones within VR environments.

2.1 Indirect Touch

Indirect touch enables comfortable input without requiring users to look at the input surface [2, 22], making it suitable for a variety of situations. For example, a trackpad or a pen tablet [22, 25, 26, 28] can be used to input data on a computer while watching a display constitutes an indirect touch scenario. In addition, indirect touch is being explored in scenarios in which mobile devices, such as smartphones and smartwatches, serve as input surfaces for interacting with smart TVs [29] or virtual windows in VR environments [24].

Because precise touch-down gestures are difficult to perform in indirect touch scenarios [28], researchers have proposed tap gestures specifically designed for indirect touch interaction [1, 13, 15, 19, 27]. In this approach, a pointer indicating the user's finger position is displayed on the virtual screen, and the user moves their finger to the target position while referencing the pointer. When the user then performs a selection action (e.g., a touch-up [1, 27] or a press input [13, 15]), the target at that location is selected. Although this gesture is slower than direct tapping, it enables the selection of small targets with a lower error rate [19].

However, gestures other than taps have received little attention in indirect touch scenarios. Wada et al. [27] designed novel gestures for indirect touch by extending this pointer-based tap selection approach. In Japanese text entry, characters are entered by performing a swipe in a specific direction after touching down on a target key. Wada et al. reversed this input procedure and proposed gestures in which users first perform a swipe gesture and then select the target key using an indirect tap triggered by a touch-up event. We considered that this reversal of the input sequence could be applied to design gestures such as double-tap and long-press for indirect touch scenarios.

2.2 Smartphone Interaction in VR Environments

Previous studies have investigated optimal interaction methods for smartphone use within VR environments [31]. The interaction methods examined include smartphone-based touch input [6, 17, 18], hand-based touch input [9], and ray-casting methods [30]. In smartphone-based and hand-based touch input, the virtual screen is anchored to the smartphone or the user's hand, respectively, whereas in ray-casting, interaction is enabled without directly touching a physical surface. Among these methods, interacting by touching the smartphone resulted in the highest selection performance, which was attributed to the tactile feedback and proprioceptive cues provided by the physical device [31]. These findings suggest the usefulness of interaction scenarios in which the smartphone itself is used to operate the smartphone.

Meanwhile, in indirect touch scenarios, directly touching the smartphone is not suitable because precise input is difficult, and existing implementations of indirect touch gestures for smartphone interaction are limited. In such scenarios, either ray-casting methods are employed [30, 31], or users must directly observe the device displayed in the VR environment for each input [9, 30]. Although tap gestures have been proposed for indirect touch scenarios, existing systems lack support for other essential gestures, such as double-tap and long-press. In this work, we designed indirect touch gestures necessary for smartphone interaction, aiming to enable comfortable and effective indirect interaction with smartphones.

3 Indirect Touch Gestures for Smartphone Interaction

We implemented a set of indirect touch gestures for smartphone interaction, which includes tap, double-tap, long-press, drag, and

[1] https://github.com/inaniwaudon/smartphone-indirect-touch-gestures

scroll gestures. Previous studies have proposed tap gestures for indirect touch scenarios that rely on touch-up events rather than touch-down events [1, 27]. However, research has not fully explored how to integrate gestures beyond tap into indirect touch interaction.

Building on the approach of reversing the input procedure for indirect touch scenarios [27], we implemented double-tap and long-press gestures for precise target selection, in which the gestures are executed before moving the finger toward the target. Because drag gestures are used to move an object after it has been selected, we implemented drag by selecting the object via a holding gesture after moving the finger to the target. Finally, because indirect tap gestures share similar input procedures with scroll gestures, directly applying scroll gestures to indirect touch scenarios is challenging. We focused on the observation that velocity profiles differ between scroll gestures and pointing gestures [10, 20, 21], and therefore hypothesized that scroll gestures could be identified based on users' finger movements.

The five gestures are classified using the following procedure: (1) If a double-tap is performed immediately after the user's finger touches down, the input is classified as a double-tap gesture. (2) If the user's finger remains pressed beyond a predefined holding threshold for long-press gestures after touch-down, the input is classified as a long-press gesture. (3) If the classification model identifies a scroll gesture based on the user's finger movement, the input is classified as a scroll gesture. (4) After the finger has moved, if it is held beyond the predefined holding threshold for drag gestures, the input is classified as a drag gesture; otherwise, it is classified as a tap gesture.

To utilize these gestures, the following parameters and model are required: (1) a holding threshold for long-press gestures, (2) a holding threshold for drag gestures, and (3) a classification model for tap and scroll gestures. The optimal values for these thresholds and the model were determined through the pilot study described below.

4 Pilot Study

We conducted a pilot study to investigate the optimal parameters for our indirect touch gestures and to collect data for developing a classification model for tap and scroll gestures. The study involved eight participants (eight males; mean age: 22.75 years, SD = 1.09 years) recruited from the authors' laboratory. All participants were right-handed. Each experimental session lasted approximately 30 minutes.

Our system consisted of a smartphone (iPhone 14), an HMD (Meta Quest 3), and a laptop PC (MacBook Air, M1, 2020). The smartphone featured a touchscreen with a resolution of 1170 × 2400 px and a pixel density of 326 ppi. We developed a unified web application using TypeScript and React (ver. 19.1.0) that ran on both the smartphone and the HMD. Interaction events were synchronized in real time via a WebSocket server running on the PC.

4.1 Task and Procedure

The pilot study consisted of two subtasks—a pointing task and a scrolling task—conducted in an indirect touch scenario within a VR environment. The smartphone screen was anchored in space at a depth of 1.5 m in front of the participant. The screen size was 20° width and 41° height.

4.1.1 Pointing Task. This task required participants to select a target rectangle using a tap gesture and was adapted from a previous study [7]. On the smartphone screen, 72 rectangles were randomly displayed within a grid of 6 × 12 cells. The target rectangle was colored pink, while the remaining rectangles were blue. The rectangle size was randomly set to either 66 px or 132 px, and one rectangle was randomly designated as the target. A yellow circular pointer indicated the participant's finger position; when the pointer hovered over a rectangle, that rectangle turned green. When the target was selected via the tap gesture, the trial ended, and another rectangle was randomly selected as the next target.

4.1.2 Scrolling Task. This task required participants to select a target row using a scroll gesture and was adapted from a previous study [31]. The smartphone screen displayed a list of row elements, each containing a number and having a height of 240 px. Ten elements were visible on the screen at a time, and the list could be scrolled vertically. A white semitransparent overlay with a height 1.5 times that of a row element was displayed at the center of the screen. A trial was considered successful when the target row stopped within the overlay. For the scroll gesture, we implemented inertial scrolling[2] to simulate standard smartphone interactions and allowed participants to stop the scrolling by touching the screen.

Participants were required to select the row element corresponding to the target number displayed in the upper left corner of the screen. The target numbers were positioned 5, 10, 20, or 30 rows away from the starting row element. Each participant completed 25 repetitions for each of the four distance conditions, which were presented in the fixed order of 5, 10, 20, and 30.

4.1.3 Procedure. The participants sat on chairs in front of a desk in a room at our university. We explained the tasks, obtained informed consent, and asked the participants to complete a prestudy questionnaire. The participants then wore the HMD and held the smartphone in their dominant hand. Throughout the study, participants were instructed to perform all tasks using only the thumb of their dominant hand. The order of the tasks was counterbalanced, and participants took breaks of at least two minutes between tasks. Before each task, participants completed 20 practice trials. Each task consisted of 100 trials in total.

4.2 Results

We collected data from 100 trials per participant for both the pointing and scrolling tasks. One trial from the pointing task and one trial from the scrolling task were excluded from the analysis because the participant misunderstood the input gesture. From the trial data, we extracted finger trajectories recorded during gesture execution. A data point corresponding to a single gesture was defined as the interval from when the finger touched the surface to when it was lifted, and all gestures performed within a single trial were extracted. In this dataset, gestures with execution times of 5 s or longer were removed as outliers (tap: 2, scroll: 8) because in these instances, participants failed to release their finger from

[2]https://developer.oculus.com/documentation/unity/unity-isdk-interaction-sdk-overview/

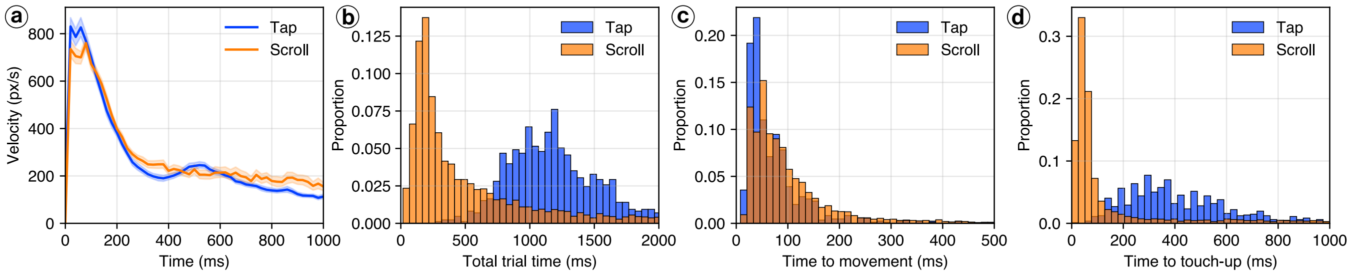

Figure 2: Results of (a) velocity profiles and histograms of (b) total trial time, (c) time to movement, and (d) time to touch-up for tap and scroll gestures.

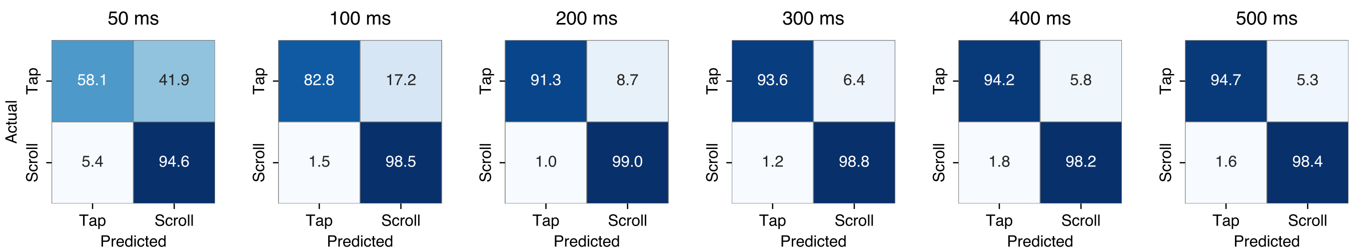

Figure 3: Row-normalized confusion matrices (in percentages) comparing actual and predicted gesture classes across different window sizes (50, 100, 200, 300, 400, and 500 ms).

the screen or exhibited unnatural behaviors, which may not reflect standard gesture execution. In addition, one data point from the pointing task was removed as a clear outlier (execution time: 90 ms). As a result, we obtained 862 tap gestures from the pointing task and 4,700 scroll gestures from the scrolling task.

4.2.1 Statistical Analysis. Fig. 2 shows the velocity profiles, gesture execution times (total trial time), the time from finger touch-down to the onset of finger movement exceeding 20 px (time to movement), and the time from when the finger has moved 20 px to the onset of finger touch-up (time to touch-up) for tap and scroll gestures. For tap gestures, the 95th percentile of the time to movement was 207.45 ms, and the 95th percentile of the time to touch-up was 941.00 ms. To prioritize the reliability of frequent tap interactions, we defined the holding threshold distinguishing tap and long-press gestures as 210 ms and the threshold distinguishing tap and drag gestures as 950 ms. These thresholds ensured that 95% of intended taps were not misclassified as long-press or drag gestures.

4.2.2 Gesture Classification Model. We developed a classification model for tap and scroll gestures using data from the first 50, 100, 200, 300, 400, and 500 ms after touch-down. We used LightGBM [16] as the classification model and split the dataset as follows: 80% for training and 20% for testing. We evaluated the model using 5-fold cross-validation, and calculated the accuracy and AUC-ROC.

We used a total of 13 features as input features for the model: three velocity-related features (standard deviation, coefficient of variation, and the 80th percentile value), two acceleration-related features (kurtosis and skewness), three jerk-related features (kurtosis, maximum value, and the time ratio until the maximum acceleration is reached), the deceleration ratio, the 40th and 80th percentile values of the finger trajectory distance, the straightness of the finger trajectory (defined as the distance between the starting point and the endpoint divided by the total trajectory length), and a binary indicator of whether the gesture was completed within the time window. These features were selected because they contributed strongly to the model's performance within a 200 ms window.

Fig. 3 shows the row-normalized confusion matrices. The model performance for each window size was as follows: an accuracy of 88.9% (SD = 1.0%) with an AUC-ROC of 0.878 at 50 ms, an accuracy of 96.0% (SD = 0.4%) with an AUC-ROC of 0.979 at 100 ms, an accuracy of 97.8% (SD = 0.6%) with an AUC-ROC of 0.997 at 200 ms, an accuracy of 98.0% (SD = 0.6%) with an AUC-ROC of 0.999 at 300 ms, an accuracy of 97.6% (SD = 0.7%) with an AUC-ROC of 0.999 at 400 ms, and an accuracy of 97.8% (SD = 0.3%) with an AUC-ROC of 0.999 at 500 ms. These results indicate that using finger trajectory data spanning 200 ms or longer enables highly accurate classification of tap and scroll gestures.

5 Discussion

We discuss the validity of the determined thresholds, the trade-off between classification accuracy and system latency, and the limitations of this work.

The holding threshold distinguishing tap and long-press gestures and the threshold distinguishing tap and drag gestures may require further adjustment for practical use. Based on the results of the pointing task, we determined that the threshold distinguishing tap and long-press gestures is 210 ms; however, considering the default threshold for long-press gestures on iOS is 0.5 s [4], we recommend using 0.5 s as the initial value even in indirect touch scenarios to support intuitive user interaction. Furthermore, the threshold distinguishing tap and drag gestures was determined to be 950 ms, which indicates that participants spent a relatively long time carefully adjusting their finger position before touch-up. Because users typically confirm that the pointer is hovering over the target before performing a touch-up, the time to touch-up is expected to be longer for smaller targets. Therefore, when drag gestures are used for tasks such as copying small text, a conservative threshold (e.g., 1.5 s) should be adopted.

While our model achieved high classification accuracy, there is a trade-off between performance and system latency. Our model accurately classified tap and scroll gestures even with a 200 ms window (accuracy: 97.8%, AUC-ROC: 0.997), and the model using a 300 ms window achieved the highest accuracy (accuracy: 98.0%, AUC-ROC: 0.999). Although higher accuracy is desirable, larger window sizes introduce additional latency, highlighting the need to identify parameters that balance classification performance and usability.

This work has several limitations regarding its generalizability and the experimental environment. First, the model is not designed to recognize horizontal swipes; thus, further data collection is needed to incorporate these gestures and expand the gesture vocabulary. Second, the participant pool was limited to eight right-handed male participants. Evaluation across a more diverse demographic is required to develop a more robust and generalizable model. Finally, the latency of our WebSocket-based implementation could potentially have influenced the execution times; optimal thresholds and models may vary depending on the level of latency.

6 Conclusion

In this work, we present a set of indirect touch gestures for smartphone interaction, which includes tap, double-tap, long-press, drag, and scroll gestures. We determined that the holding threshold distinguishing tap and long-press gestures and the threshold distinguishing tap and drag gestures were 210 ms and 950 ms, respectively. In addition, we developed a model that classifies tap and scroll gestures with an AUC-ROC of 0.997 using data from the first 200 ms after touch-down. These contributions expand the range of input techniques available for smartphone interaction in indirect touch scenarios.

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Number 24K21319.

References

[1] Sunggeun Ahn, Seongkook Heo, and Geehyuk Lee. 2017. Typing on a Smartwatch for Smart Glasses. In *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces* (Brighton, United Kingdom) *(ISS '17)*. Association for Computing Machinery, New York, NY, USA, 201–209. doi:10.1145/3132272.3134136

[2] Diogo Almeida, Daniel Mendes, and Rui Rodrigues. 2023. SIT6: Indirect Touch-Based Object Manipulation for DeskVR. *Computers & Graphics* 117 (2023), 51–60. doi:10.1016/j.cag.2023.10.013

[3] Apple. 2025. *Human Interface Guidelines.* Retrieved January 23, 2025 from https://developer.apple.com/design/human-interface-guidelines

[4] Apple. 2025. *init(minimumDuration:) | Apple Developer Documentation.* Retrieved January 4, 2025 from https://developer.apple.com/documentation/swiftui/longpressgesture/init(minimumduration:)

[5] Apple. 2025. *Use View Mirroring or AirPlay Receiver on Apple Vision Pro.* Retrieved January 23, 2025 from https://support.apple.com/en-us/guide/apple-vision-pro/tanf4ca9ada2/visionos

[6] Huidong Bai, Li Zhang, Jing Yang, and Mark Billinghurst. 2021. Bringing Full-Featured Mobile Phone Interaction into Virtual Reality. *Computers & Graphics* 97 (2021), 42–53. doi:10.1016/j.cag.2021.04.004

[7] Zhuojiang Cai, Jingkai Hong, Zhimin Wang, and Feng Lu. 2025. GazeSwipe: Enhancing Mobile Touchscreen Reachability through Seamless Gaze and Finger-Swipe Integration. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 1107, 14 pages. doi:10.1145/3706598.3713739

[8] Camille Dupré, Caroline Appert, Stéphanie Rey, Houssem Saidi, and Emmanuel Pietriga. 2024. TriPad: Touch Input in AR on Ordinary Surfaces with Hand Tracking Only. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 754, 18 pages. doi:10.1145/3613904.3642323

[9] Camille Dupré, Emmanuel Pietriga, Olivier Gladin, Stéphanie Rey, Houssem Saidi, and Caroline Appert. 2025. Investigating Hand-Bound Pads for AR Input Using Hand-Tracking Only. *Proceedings of the ACM on Human-Computer Interaction* 9, 5, Article MHCI018 (Sept. 2025), 25 pages. doi:10.1145/3743707

[10] Kenneth Flowers. 1975. Ballistic and Corrective Movements on an Aiming Task. *Neurology* 25, 5 (1975), 413–413. doi:10.1212/WNL.25.5.413

[11] Google. 2025. *Material Design 3.* Retrieved January 23, 2025 from https://m3.material.io

[12] Lucas Hartman, Ethan Pigou, Nicholas Strzelczyk, Santiago Gomez-Rosero, and Miriam A. M. Capretz. 2025. Robust Smartphone Screen Integration with Deep Learning for Virtual Reality Pass-through. In *2025 IEEE 49th Annual Computers, Software, and Applications Conference* (Toronto, ON, Canada) *(COMPSAC 2025)*. IEEE Computer Society, Los Alamitos, CA, USA, 1437–1443. doi:10.1109/COMPSAC65507.2025.00180

[13] Jay Henderson, Jessy Ceha, and Edward Lank. 2020. STAT: Subtle Typing Around the Thigh for Head-Mounted Displays. In *Proceedings of the 22nd International Conference on Human-Computer Interaction with Mobile Devices and Services* (Oldenburg, Germany) *(MobileHCI '20)*. Association for Computing Machinery, New York, NY, USA, Article 27, 11 pages. doi:10.1145/3379503.3403549

[14] Immersed. 2025. *Immersed.* Retrieved January 23, 2025 from https://immersed.com

[15] Haiyan Jiang and Dongdong Weng. 2020. HiPad: Text entry for Head-Mounted Displays Using Circular Touchpad. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces* (Atlanta, USA) *(IEEE VR 2020)*. IEEE Computer Society, Los Alamitos, CA, USA, 692–703. doi:10.1109/VR46266.2020.00092

[16] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS '17)*. Curran Associates Inc., Red Hook, NY, USA, 3149–3157. doi:10.5555/3294996.3295074

[17] Stanislav Kyian and Robert Teather. 2021. Selection Performance Using a Smartphone in VR with Redirected Input. In *Proceedings of the 2021 ACM Symposium on Spatial User Interaction* (Virtual Event, USA) *(SUI '21)*. Association for Computing Machinery, New York, NY, USA, Article 6, 12 pages. doi:10.1145/3485279.3485292

[18] Akhmajon Makhsadov, Donald Degraen, André Zenner, Felix Kosmalla, Kamila Mushkina, and Antonio Krüger. 2022. VRySmart: a Framework for Embedding Smart Devices in Virtual Reality. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 358, 8 pages. doi:10.1145/3491101.3519717

[19] Fabrice Matulic, Aditya Ganeshan, Hiroshi Fujiwara, and Daniel Vogel. 2021. Phonetroller: Visual Representations of Fingers for Precise Touch Input with Mobile Phones in VR. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 129, 13 pages. doi:10.1145/3411764.3445583

[20] David E. Meyer, Richard A. Abrams, Sylvan Kornblum, Charles E. Wright, and J. E. Keith Smith. 1988. Optimality in Human Motor Performance: Ideal Control of Rapid Aimed Movements. *Psychological Review* 95, 3 (1988), 340–370. doi:10.1037/0033-295x.95.3.340

[21] Takahiro Nishimura, Kouki Doi, and Hiroshi Fujimoto. 2021. Ballistic and Corrective Movements of Drag on Touch Screens. *CMBES Proceedings* 44 (May 2021), 1 pages. https://proceedings.cmbes.ca/index.php/proceedings/article/view/922

[22] Dominik Schmidt, Florian Block, and Hans Gellersen. 2009. A Comparison of Direct and Indirect Multi-touch Input for Large Surfaces. In *Human-Computer Interaction – INTERACT 2009*. Springer Berlin Heidelberg, Berlin, Heidelberg, 582–594. doi:10.1007/978-3-642-03655-2_65

[23] Han Shi, Hanzhong Luo, HyeonBeom Yi, and Seungwoo Je. 2025. ReachPad: Interacting with Multiple Virtual Screens using a Single Physical Pad through Haptic Retargeting. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 1135, 17 pages. doi:10.1145/3706598.3713629

[24] Prabhakar V. Vemavarapu and Christoph W. Borst. 2017. Indirect Touch Interaction with Stereoscopic Displays using a Two-Sided Handheld Touch Device. In *2017 IEEE Symposium on 3D User Interfaces* (Los Angeles, CA, USA) *(3DUI 2017)*. IEEE Computer Society, Los Alamitos, CA, USA, 209–210. doi:10.1109/3DUI.2017.7893345

[25] Simon Voelker, Andrii Matviienko, Johannes Schöning, and Jan Borchers. 2015. Combining Direct and Indirect Touch Input for Interactive Workspaces using Gaze Input. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction* (Los Angeles, California, USA) *(SUI '15)*. Association for Computing Machinery, New York, NY, USA, 79–88. doi:10.1145/2788940.2788949

[26] Simon Voelker, Chat Wacharamanotham, and Jan Borchers. 2013. An Evaluation of State Switching Methods for Indirect Touch Systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 745–754. doi:10.1145/2470654.2470759

[27] Yuto Wada, Myungguen Choi, Kaoru Shirane, and Buntarou Shizuki. 2025. Flick-in: Japanese Text Entry Method for Indirect Touch Using Bezel-Initiated Swipe. In *Proceedings of the 2025 ACM Symposium on Spatial User Interaction* (Montreal, QC, Canada) *(SUI '25)*. Association for Computing Machinery, New York, NY, USA, Article 19, 13 pages. doi:10.1145/3694907.3765927

[28] Yizhong Xin, Ruonan Liu, and Yan Li. 2020. Strategy for Improving Target Selection Accuracy in Indirect Touch Input. *IEICE Transactions on Information and Systems* E103.D, 7 (2020), 1703–1709. doi:10.1587/transinf.2019EDP7218

[29] Zhican Yang, Chun Yu, Xin Yi, and Yuanchun Shi. 2019. Investigating Gesture Typing for Indirect Touch. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3, Article 117 (Sept. 2019), 22 pages. doi:10.1145/3351275

[30] Fengyuan Zhu, Xun Qian, Daniel Kalmar, Mahdi Tayarani, Eric J. Gonzalez, Mar Gonzalez-Franco, David Kim, and Ruofei Du. 2025. Beyond the Phone: Exploring Phone-XR Integration through Multi-View Transitions for Real-World Applications. In *2025 IEEE Conference Virtual Reality and 3D User Interfaces* (Saint-Malo, France) *(IEEE VR 2025)*. IEEE Computer Society, Los Alamitos, CA, USA, 770–780. doi:10.1109/VR59515.2025.00099

[31] Fengyuan Zhu, Mauricio Sousa, Ludwig Sidenmark, and Tovi Grossman. 2024. PhoneInVR: An Evaluation of Spatial Anchoring and Interaction Techniques for Smartphone Usage in Virtual Reality. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 639, 16 pages. doi:10.1145/3613904.3642582