2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)

Touching The Droid: Understanding and Improving Touch Precision With Mobile Devices in Virtual Reality

Fengyuan Zhu*
Department of Computer Science, University of Toronto

Zhuoyue Lyu †
Department of Computer Science, University of Toronto

Mauricio Sousa‡
Department of Computer Science, University of Toronto

Tovi Grossman§
Department of Computer Science, University of Toronto

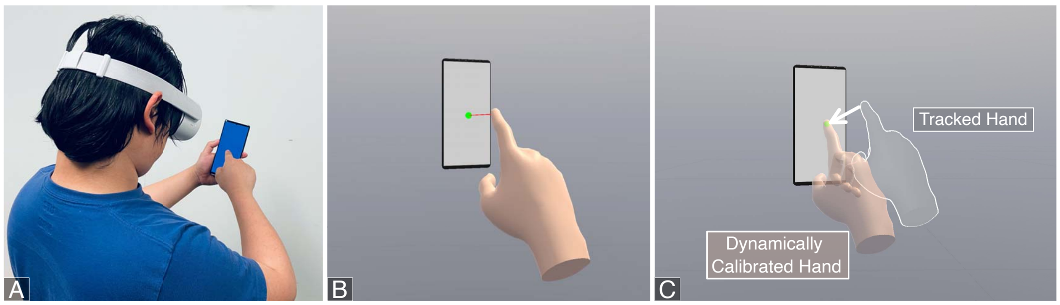

Figure 1: In this work, we improve interaction with touch-based devices in VR by contributing an approach to reduce the imprecisions in the alignment of the visual representation of either the physical hands or device that can impact the precision of touch. A) Before a user performs a touch interaction on a device tracked by a Head-Mounted Display, B) our Fingertip Estimating Algorithm will predict the fingertip position and expected touch position in the virtual world, and C) Once the touch occurs, our Dynamic Calibration Algorithm retargets the position Tracked Hand into a new corrected position.

Abstract

Touch interaction with physical smartphones and tablets in Virtual Reality offers interesting opportunities for cross-device input. Unfortunately, any imprecision in the alignment of the visual representation of either the hand or device can impact the precision of touch and the realism of the experience. We first study a user's ability to rely solely on preoperative feedback to perform touch interaction in VR, where no rendering of the hand is provided. Results indicate that touch in VR is possible without a visual representation of the hand, but accuracy is influenced by how the device is held and the distance traveled to the target. We then introduce a dynamic calibration algorithm to minimize the offset between the physical hand and its virtual representation. In a second study, we show that this algorithm can increase touch accuracy by 43%, and minimize depth-based "screen penetration" or "floating touch" errors.

Index Terms: Human-Centered Computing—Human computer interaction (HCI)—Interaction Paradigms—Virtual Reality

1 Introduction

Modern Head-mounted Displays (HMDs) completely block awareness of the real world to allow engaging and immersive Virtual Reality (VR) experiences (Figure 1A). Nevertheless, using physical touch devices, such as smartphones and tablet devices, as input for VR is trending because they are reliable and accurate. Previous research suggests that VR users can benefit from the affordances and the familiarity provided by touch-based input interactions [34, 44, 52].

Visualizing touch interactions in virtual environments requires tracking people's hands and touch input devices in 3D. To enable touch-based input in VR, existing research have employed various tracking approaches including commercial external optical tracking of reflective markers, such as Vicon[1] or Optitrack[2], head-mounted depth cameras like Leap-motion[3]or built-in cameras, or hand segmentation projection via augmented video see-through techniques [27, 33, 34]. Nonetheless, these approaches do not deliver the adequate millimetric (*mm*) accuracy needed for precise touch interactions [25].

A perceptive miss-alignment between physical hands and virtual representations can impact touch input's precision and the experience's realism. A study by Schneider et al. [40, 41] revealed that most of the existing tracking methods could have a *cm* level of offset when a touch occurs. Furthermore, VR touch errors are not just limited to the 2D screen space, as z-depth errors perpendicular to the touch surface can also occur. Errors such as *"Screen Penetration"* (Figure 2A) or *"Floating Touch"* (Figure 2B) can impair the consistency between visual and haptic feedback [12], confuse users and hinder the experience. Additionally, the newer inside-out tracking techniques used in modern HMDs may increase the risk of z-depth errors. Touch motions are most likely perpendicular to the camera plane, resulting in more frequent unexpected errors.

In this work, we aim to eliminate those offsets and improve the precision of touch-based interactions in VR by correcting the alignment of the visual representation. For this, we first evaluate whether VR users can rely solely on proprioceptive feedback to perform touch interaction, eliminating visual offset errors: we wish to understand how well users could perform touch interactions without visual hand representation. An essential factor is the usage of "body-relative" locations as a real-world physical reference [37]. Thus

*e-mail: fyzhu@dgp.toronto.edu
†e-mail: zhuoyue.lyu@mail.utoronto.ca
‡e-mail: mauricio@dgp.toronto.edu
§e-mail: tovi@dgp.toronto.edu

[1] Vicon: https://www.vicon.com
[2] OptiTrack: https://optitrack.com
[3] Leap Motion Controller: https://www.ultraleap.com/product/leap-motion-controller/

978-1-6654-5325-7/22/$31.00 ©2022 IEEE
DOI 10.1109/ISMAR55827.2022.00099

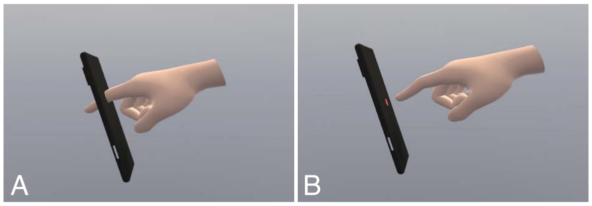

Figure 2: Types of screen depth errors from Chan et al. [12]: A) *Screen Penetration*, and B) *Floating Touch*.

we designed the first study with three phone-holding postures: 1) holding the phone and tapping with their thumb on the same hand; 2) holding the phone and tapping with the opposite hand's index finger; And 3) the phone is mounted and accessed with the user's index finger. The results showed that for continuous On-Screen touches, participants achieved an accuracy of (6.84 ± 0.49 mm) regardless of their posture. Yet, for the "first off-screen touch," the accuracy significantly increases with greater spatial reference (Thumb: 6.10 ± 0.49mm, Right Index: 7.9 ± 0.37mm, Table Mounted: 10.1 ± 0.63mm).

No matter how accurate proprioceptive touch can be, it is ideal for providing a visualization of the user's hand when touching in VR. Inspired by previous dynamic calibration methods for gaze pointing [17], we noticed that we could take advantage of the "touch event" to eliminate tracking errors. Once a touch occurs, we can calculate the position of the "virtual touch" in the 3D virtual environment based on the 2D touch data retrieved from the touch device. By aligning this virtual touch position with the virtual fingertip, we can dynamically calibrate the relative position of the touch device to the user's finger. This alignment could reduce the relative errors between the virtual and real world. Also, based on previous work [25], we employ a dynamic fingertip estimation algorithm combining the user's intention and geometry relationship between hand and device, as demonstrated in Figure 1B and C.

We conducted a second study to evaluate the effectiveness of our dynamic calibration and fingertip estimation approach. The results showed that our dynamic calibration algorithm improves the virtual touch accuracy by 43%, and reduces z-depth penetrating and floating touch errors.

Therefore, in this paper, we contribute: 1) results from a first user study suggesting the performance of touch input in VR is independent of user posture and virtual hand representation; 2) a dynamic calibration algorithm that minimizes the positional offset between the user's physical hand and its virtual representation; and 3) results from the user study suggesting that our method improves accuracy and minimizes depth-based errors.

2 Related Work

This work builds upon precious research on touch interaction in 2D, haptic feedback for virtual target acquisition, and combining 2D devices with HMDs.

2.1 Touch Interaction in 2D

Numerous works have contributed improvements to touch accuracy in 2D. One way to improve accuracy is to build better models that fit the data. For example, based on Fitts' original studies in 1954 [18], Bi et al. [8] derived the FFitts model for finger touch input. Holz and Baudisch found "parallax" artifacts caused the inaccuracy and introduced the projected center model [25].

Another approach to improve touch accuracy is to use alternative forms of interaction, such as Back-of-Device (BoD) touch interactions [7, 24, 48, 49] which avoid the occlusion caused by the finger when directly touching the screen [42]. Xia et al. [50] introduced NanoStylus, a finger-mounted stylus for high precision pointing capability on ultra-small devices. These works inspire us to use alternative finger representation or techniques for touch input in VR environments.

2.2 Haptic Feedback for Virtual Target Acquisition

Virtual target acquisition can be challenging due to the lack of haptic feedback. Chan et al. [12], and Lubos et al. [32] found that people perform poorly in determining the z-coordinate (depth, viewing axis) of spatial targets. Yet, a physical surface or visual guidance cues can counteract this targeting issue [3, 15]. Other strategies include showing a visual surface and a target curve [3], using a physical surface to enhance touching virtual objects [6,51] providing audio and pseudo-shadow feedback [15], wearing a haptic device to simulate various physical buttons [26], using self-haptics and retargeting to provide tactile with virtual objects [16], and using an ultrasound transducer array to create tactile focal points in mid-air [11, 38]. However, the most straightforward approach may be holding a physical object in hand, which provides real-world haptic feedback, and the user is presented with more stimuli, thus providing higher fidelity [30]. Since the object resides closely to the body, users can take advantage of the proprioception sense that sometimes works pretty well even without visual feedback [49]. This previous research shows that the target acquisition in VR can be more effective if the virtual experience provides haptic feedback.

2.3 Combining 2D Devices and HMDs

By combining touch devices and HMDs, researchers have explored numerous scenarios that cross the boundary between the screen to the spatial environment, either taking usage of the users' body frame [1], around touching devices [22, 39] and in the spatial environment [10, 52]. Previous works also explored the benefits of using tablets [2, 14, 44] or smartphones [36] to provide precise input for VR/AR applications. Several works also focus on allowing smartphones in virtual environments [5, 27, 33, 34].

In addition to developing interactive systems, previous research works have investigated the target acquisition performance in the context of using the touch screen in VR. Valkov et al. [46] evaluated depth-sensing performance with the stereo display system when touching a screen. Biener et al. [10] evaluated the user's capability to interact with information that reaches beyond a single physical touchscreen in the virtual spatial environment. Matulic et al. [34] use the augmented video segmentation to visualize fingers over the phone and evaluate the touch performance based on the tech. Son et al. [43] designed an approach to estimate thumb touch position and evaluated the typing performance with two thumbs in VR. Recently, Schneider et al. [40, 41] compared touch accuracy with multiple consumer-ready techniques that provide hand tracking in VR/AR. Their research found that most tracking solutions have inevitable errors in the z-direction when a physical touch occurs.

These works studied the touch accuracy with the common usage of 2D touch devices in VR with multiple tracking solutions. Few research works have tried only to render the touch device without the users' hand. Yet, no one has proposed an effective method to reduce these errors significantly – especially in the z-direction when the touch occurs. Together, this collection of prior work motivates us to fill these gaps in the existing body of research.

3 Study 1: Proprioception Study without Rendering User's Hand

While hand tracking is becoming more common in VR headsets, it is still not native across all headset technologies. In its absence, it may not be possible to visualize the hand location accurately. Furthermore, even when hand tracking is available, there is the inevitable tracking inaccuracy of the fingertips' location [40, 41].

We hypothesize that one way to eliminate the offset of the visual hand location may not be to render it at all.

We conducted this first study to assess the participant's ability to select touch-based input targets in VR without visualizing their one hands and fingers. We asked participants to rely on their sense of proprioception for target selection on a virtual proxy of a physical touch device. An essential factor for proprioceptive feedback is "body-relative" interaction – where there is a real-world physical frame of reference [37]. We employed three common holding postures from previous works [19,29], to represent different levels of "body-spatial relative frame"(Figure 3): 1) *Index*, participants would hold the phone with one hand and tap with the opposite hand's index finger (medium spatial reference); 2) *Mounted*, the phone was mounted on a fixed tripod and tapped with the index finger (least significant spatial reference); and 3) *Thumb*, where participants would hold the phone with one hand and tap with their thumb (most significant spatial reference).

3.1 Study Design

The independent variables of the study are the three different *Postures*; the *Distance* between targets (small: 26.1 mm, medium: 34.8 mm, large: 52.2 mm); and *Size*, the diameter of the target (small: 2.4 mm, medium: 4.8 mm, large: 7.2 mm). As dependent variables, we used *Accuracy* (represented by Distance-to-target, the Euclidean distance between the touch point to the center of the target) and *Time* (the elapsed time between a participant being presented with a target and their finger touching the screen).

Inspired by Bi et al. [8, 9], our study consisted of both on-screen-start and off-screen-start target acquisition tasks that relate to the position of the finger when the trial starts is within the screen area or not. To achieve the off-screen-start target acquisition condition, we instructed the participants to click a mouse within their arm's length range but apart from the phone. For the on-screen-start target acquisition condition, participants continuously acquired targets using the same finger and remained in the nearby area of the screen.

We designed a within-subjects study with 3 x 3 x 3 (Posture x Size x Distance) conditions. We use a Balanced Latin Square to counter-balance Posture, and the nine combinations of Size X Distance were randomized within each posture. We generated three blocks containing 12 trials as a group for each condition. The 12 trials contained one off-screen-start trial and 11 on-screen-start trials. The participants saw a notification to start an off-screen operation once they finished the 11 on-screen trails.

In total, there were 3 locations x 9 conditions x 3 blocks x 12 trials (1 off-screen + 11 on-screen) = 972 trials per participant.

3.2 Apparatus and Set-Ups

We used a Google Pixel 2 smartphone (5.0 inches, 1080 x 1920 pixels display with 441 pixels per inch) and an Oculus Rift S. Similarly to Bai et al. [5], we employed an Oculus Touch Left Hand Controller attached to the phone using a custom 3D printed holder to track its position in the real world. We rendered a virtual representation of the phone in the VR environment.

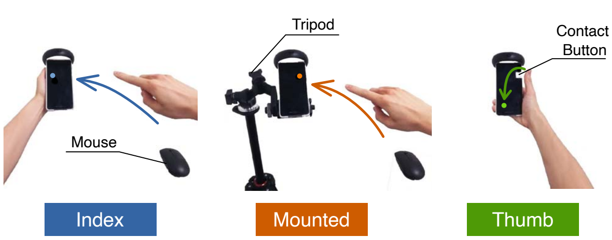

Figure 3: Evaluation conditions: Index, Mounted, and Thumb employed in Study 1.

To simulate the off-screen-start target acquisition tasks, we fixed a mouse on a table to trigger a new block of trials (for Index and Mounted conditions). For the Thumb condition, we instructed participants to hold and operate the phone with the same hand, so we added a piece of aluminum foil as a contact button (15mm x 40mm) to the top-right corner of the phone.

We developed the evaluation prototype in Unity 2019 LTS on a PC with a GTX 1080 6GB and an Intel(R) Xeon(R) CPU E5-2630 with 64 GB as physical memory. The phone was connected to the PC using a customized WiFi 6 network to transfer the study's touch data via UDP.

To control the relative position of the phone in 3D space and the participants holding postures, we built two visual widgets in the virtual environment, a translucent plane and a translucent bounding green sphere. The translucent plane, situated perpendicular to the phone screen, appears and turns red when the participant is not facing squarely (over 10°) at the screen. The translucent bounding green sphere (diameter = 26 cm) surrounded the smartphone and turns red when the smartphone was out of the sphere bounds, or when the participant was too close to the smartphone (20% of the head-phone distance).

3.3 Participants

We recruited 12 participants (6 females) between the ages of 20 and 29 (Mean = 23.4, SD = 3.3). Two had reported having experience with VR/AR equipment, with one reporting a weekly usage exceeding 10 hours. All participants were right-handed and had normal or corrected-to-normal vision. The education level of the participants varied from undergraduate to graduate. We offered compensation to the participants.

3.4 Procedure

We instructed the participants to wear the VR headset and to adjust their body position until they were comfortable with the headset-to-phone distance. The body positions will be used to initialize the translucent plane and translucent sphere visual widgets, as a way to remind participants to maintain their original preset posture during the study. At the beginning of each *Posture* condition, we allowed participants to carry out several training sessions with randomized Size-Distance targets. The formal blocks started once they were confident with the task.

Each block started with a "Please put your finger back." instruction displayed on the virtual touchscreen for the Off-screen-start target acquisition task. Immediately after touch events, a new target was displayed. Participants were then instructed to perform On-screen-start target acquisition until all 12 targets for one group were achieved. Participants were instructed to acquire targets as quickly and accurately as possible in all trials. Participants were allowed to take a break between *Posture* conditions.

3.5 Results

We recorded data in both JSON and CSV format. We used *statsmodels* and *pandas* in python to process and analyze data. We applied a Repeated-measures analysis of variance (ANOVA-RM) for both On-Screen and Off-Screen data.

3.5.1 On-screen Accuracy

Accuracy was calculated as the euclidean distance between the touch point to the target from the phone. The first dependent variable we measured is the accuracy. The ANOVA-RM found significant effect of Different Target Distance on Accuracy ($F(2,22) = 28.370, p < .001$). The means of euclidean distance were 8.38 ± 0.76mm, 6.45 $\pm$ 0.39mm and 5.69 ± 0.37mm in order of decreasing target distance Post-hoc pairwise comparison using t-test with Holm-Bonferroni correction revealed a significant difference for Distance between medium and large ($p < .001$) and small and large ($p < .001$). There was no significant main effect for Postures and Size on Accuracy These results indicate that with the On-screen-start acquisition, different special references and target sizes have a minor impact on touch performance, while the distance between targets significantly influence the accuracy: the shorter the distance, the more accurate the target acquisition.

3.5.2 On-screen Speed

We found a significant effect on distances between targets regarding Time ($F(2,22) = 6.845, p < .001$), with means 0.74 $\pm$ 0.04s, 0.69 $\pm$ 0.04s and 0.68 $\pm$ 0.04s in order of decreasing target distance. Post-hoc pairwise comparison using t-test with Holm-Bonferroni correction revealed a significant difference in Distance between small to large ($p < .05$) , and medium to large ($p < .05$) This result indicates that similar to Accuracy, Time is significantly influenced by the distance between targets with on-screen-start target acquisition as well: the shorter the distance, the quicker the user can acquire targets.

The ANOVA-RM also showed a significant effects with different Target Size regarding Time($F(2,22) = 12.637, p < .001$), with means in 0.69 $\pm$ 0.04s, 0.70 $\pm$ 0.04s, 0.73 $\pm$ 0.04s in order of increasing target size. Post-hoc pairwise comparison using t-test with Holm-Bonferroni correction revealed significant differences in Distance between small to large ($p < 005$) and small to medium ($p < .05$) .

The results did not show a significant effect regarding Posture on Time.

3.5.3 Off-screen Accuracy & Speed

Though we did not find significant effects with on-screen postures for accuracy and speed, for the off-screen conditions, the ANOVA-RM found significant effects for Posture on both Accuracy (represented by distance to target, $F(2,22) = 25.9611, p < .001$) and Time ($F(2,22) = 12.843, p < .001$)).

The means of Accuracy for different postures were 6.10 $\pm$ 0.50mm for Thumb, 7.91 $\pm$ 0.36mm for Index, and 10.1 $\pm$ 0.63mm for the Mounted conditions (Figure 4A). A Post-hoc pairwise comparison using t-test with Holm-Bonferroni correction reveals significant differences in Postures between Index to Thumb ($p < .01$) , Mounted to Thumb ($p < .001$) and Mounted to Index ($p < .005$).

The means of Time with different postures were 1.27 $\pm$ 0.13s for Index, 1.23 $\pm$ 0.14s for Mounted and 0.75s $\pm$ 0.07s for Thumb (4B). The Post-hoc pairwise comparison using t-test with Holm-Bonferroni correction reveals significant differences between Index to Thumb ($p < .005$) and Mounted to Thumb ($p < .005$), and rejected the time difference between Mounted to Index.

The ANOVA-RM did not find statistically significance differences between Size on Accuracy and Size on Speed.

3.6 Discussion

This study confirms our hypothesis that "body relative frame" significantly influences touch target acquisition, but only with Off-screen conditions.

The non-significance of On-Screen Data in Accuracy and Speed tells us that the "body relative frame" will lose its effects once the user's hand is nearby the touch area or already interacting with the phone.

We expected the significance of on-screen behavior regarding Target Distance and Size on Accuracy and Speed since this is typical behavior of direct manipulations. The post-hoc comparison results also support this.

Figure 4 illustrates the contrasting results for accuracy between the On-Scren and Off-Screen conditions. The overall error rates of the distance-to-target are located in an acceptable range (6.84 $\pm$ 0.49mm), which means the users could perform a relatively good touch acquisition in VR even if they can't see their hands.

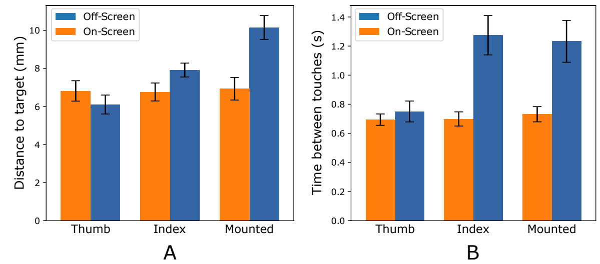

Figure 4: Results for the relationship between A) posture and distance to target, and B) Posture and time between touches.

4 Offset Correction Algorithm

The first study indicates that users can exploit their sense of proprioception to interact with touch-based devices without seeing their hands. However, accuracy did decrease for distant targets, off-screen acquisitions, and holding postures with less spatial reference.

While providing a visual representation of the hands could alleviate such accuracies, the problem of inherent fingers tracking inaccuracies [40, 41] would first need to be addressed. In this section, we introduce a novel dynamic calibration algorithm based on an understanding of the touch process [25] which could potentially improve touch accuracy in VR.

4.1 Analysis of Virtual Touch Process

Interactions in VR are different from the physical world. In VR, people rely on visual feedback to touch a virtual object. In most cases, the visual proxy representation of a physical object may not be perfectly aligned with its physical counterpart, thus creating a problematic inconsistent haptic sensation of touch. In this case, we should consider the physical instance of the mobile device, the physical instance of the user's hand, the virtual rendering of the mobile device, and the virtual rendering of the user's hand. In summary, we can describe the process of the "touch event" in VR as: 1) The users see a visual hint in VR and aim at the target relying on the relative position of their virtual hand and virtual phone. 2) Before touching the physical touch device, the users' virtual fingers **should not** touch or penetrate the virtual screen surface before feeling the haptic sensation of touching the physical touch device. 3) When users feel their real finger touching the device, the virtual touch event should be triggered. Ideally, the touch's visual feedback should coincide with their haptic feedback.

Regarding software, ideally two types of data should be recorded, one is the Physical Touch Data $TouchData_{phone} = (a_{touch}, b_{touch})$ directly retrieved from the phone, and the Virtual Touch Data $TouchPos_{virtual} = Proj_{pos} = (x_{proj}, y_{proj}, z_{proj})$ calculated based on the relative positions between the Virtual Finger and the Virtual Phone. Considering there is perfect tracking and the Virtual Position of the Finger Tip is $TipPos_{spatial} = (x_{tip}, y_{tip}, z_{tip})$, then $TipPos_{spatial}$ will be equal to $TouchPos_{virtual}$ and equal to $TouchPos_{phone}$ and defined as:

$$\begin{aligned}(x_{tip}, y_{tip}, z_{tip}) &== (x_{proj}, y_{proj}, z_{proj}) \\ &== (x_{touch}, y_{touch}, z_{touch}) \\ &== \frac{(a_{touch}, b_{touch}, 0)}{ppcm} \mathbf{M}_{PhoneToSpace}\end{aligned}$$

Where $(x_{tip}, y_{tip}, z_{tip})$ refers to the position of the virtual fingertip, $ppcm$ refers to the Pixel Per Centimeter of the screen resolution, and $\mathbf{M}_{PhoneToSpace}$ is the Linear Matrix that turns the point from the Virtual Phone Space into the Virtual Global Space. However, in real life, most consumer-ready tracking solutions cannot accurately track consistently to a millimetric level, especially in the Z-Depth, which quickly breaks the consistency between visual and haptic [40]. Previous works applying re-targeting illusions in Virtual Reality [4, 21] show that users can tolerate a global offset applied to the visual representation as long as the haptic feedback keeps its consistency with visual feedback. Our dynamic calibration doesn't require a perfect alignment between the physical and virtual worlds. Instead, our approach needs to guarantee a consistent relative position among Virtual Finger, Virtual Phone, and Touch Point concerning the physical touch event.

4.2 Finger Tip Estimation

We consider *Fingertip* as the expected touching point on the surface of the user's virtual finger. Unlike touching the virtual user interface with virtual hands, we cannot use collision detection to estimate the point of contact between the virtual finger and the virtual phone because the virtual collision may not happen simultaneously when the physical touch occurs. A common approach is attaching a visual marker to the user's fingernail and applying a fixed delta distance to estimate the front tip of the user's finger. While this method generally works well, it assumes that participants are using the same posture when touching a screen, which is different from their day-to-day use [25, 29].

In VR, we can use real-time information to better estimate the fingertip position. Inspired by Holz et al. [25] findings on peoples' touch behaviors and how to estimate touch events, we propose a method for calculating a user's fingertip position. We describe our estimation method in Algorithm 1.

Algorithm 1 The Process of Finger Tip Estimation

Input:
Finger Key Joint Position: $Joint_{pos} = (x_{kj}, y_{kj}, z_{kj})$
Phone Position and Orientation: $Phone_{pose} = (x_p, y_p, z_p) \& \mathbf{R}_p$
Finger Geometry: $Mesh_{finger} = \{v, e, s\}$

Output:
Estimated Finger Tip Position: $Fingertip_{pos} = (x_{tip}, y_{tip}, z_{tip})$
Projected Point on the Screen: $Proj_{pos} = (x_{proj}, y_{proj}, z_{proj})$
Ray Vector from Finger Tip to Screen: $\vec{Ray}_{hover}$
Whether the projection is within boundary: $IsIn$

Process:

Calculate $Plane_{phone} = (x_p, y_p, z_p, \vec{n})$ and the boundary of valid screen area $Boundary = (a_{min}, b_{min}, a_{max}, b_{max})$, based on the $Phone_{pose}$ from the input. The plane here refers to the phone surface plane in the space, $\vec{n}$ is the normal of the plane that towards the screen.
Calculate the projected point $Proj_{space} = (x_{proj}, y_{proj}, z_{proj})$ which represent the closest point from the $Joint_{pos}$ to the $Plane_{phone}$, calculate $Proj_{local} = (a_{proj}, b_{proj}, 0)$ of the projected point in the local space of the plane.
Calculate $\vec{Ray}_{hover} = Proj_{space} - Keyjoint_{pos}$ that represent the vector from the Finger Key Joint to the Projected point
if (a_{proj}, b_{proj}) in the range of $(a_{min}, b_{min}, a_{max}, b_{max})$ **AND** $\vec{n}.\vec{Ray}_{hover} > 0$ **then**
    $IsIn =$ **TRUE**
    $Fingertip_{pos}$ = Cross Point between $\vec{Ray}_{hover}$ to $Mesh_{finger}$
    return Output List
end if

We consider the *Finger Key Joint* as a point inside the user's finger. This point represents not only the geometry center of the tip joint but also the visual clue that users would use for their targeting process [25]. Figure 5 depicts the critical point and the visual clues estimated.

Our estimation process also considers the *Projected Point* on the phone surface, marked as $Proj_{pos}$. This variable also estimates the virtual touch point when a physical touch occurs. Also, the $\vec{Ray}_{hover}$ linking the $Proj_{pos}$ and the $FingerTip$, can be used to determine whether a "floating touch" or a "penetrating" error has occurred. This estimation data is used by the Dynamic Calibration Algorithm, which we will discuss next.

4.3 Dynamic Calibration Process

Our dynamic calibration algorithm aims to ensure that the user's visual feedback maintains consistency with the haptic feedback. More precisely, this algorithm should: 1) Prevent a "Penetrating Error" when physical touch has not yet occurred; 2) Avoid a "Floating Touch Error" when physical touch takes place; 3) Make sure the virtual contact point is similar to the physical contact point when a physical touch occurs. For this, our dynamic calibration also considers the moment the finger approaches the phone before dealing with the touch event itself. We formalize our method in Algorithm 2 below.

The first part of our approach, when the physical touch has not yet occurred, reduces the virtual speed of the approaching of the virtual finger when the distance between the virtual fingertip to the virtual screen is less than the preset $Threshold$. We reduce this virtual speed to prevent penetration errors. The $Threshold$ should be set based on the tracking technology of the free hand, where we could use previous works [40, 41] as reference. The α could either be a fixed number or a dynamic float proposed by previous VR retargeting methods [4, 21].

The second part of our approach relates to when the physical touch takes place. Similarly to previous dynamic calibration techniques used for gaze pointing [17], our dynamic calibration method uses the difference between the *Virtual Fingertip, Projected Touch Point* and *Physical Touch Point* during each touch as an indicator to calculate the Δ and applies it to the hand. This could either be a one time set-up, or an accumulated process based on historical data as $\Delta =$ *WeightedOffset* $(\delta_1, \delta_2, \delta_3 \dots)$. We used one-time offset correctness for each trail.

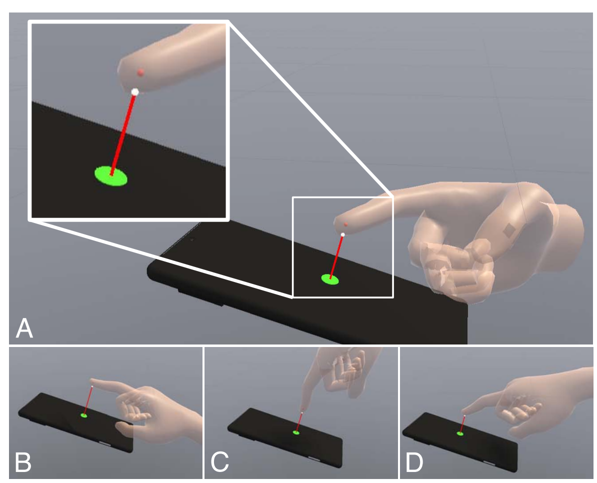

Figure 5: Finger estimation: the red point inside the finger is the $FingerKeyJoint$ mentioned within the algorithm, the white dot is the calculated results of the $Fingertip_{pos}$. The green circle is the $EstimatedTouchingPoint$ which will be used for $TouchPos_{virtual}$ when touch occurs. And from A to D we showed different postures, with the Estimated Finger Tip changed accordingly

Algorithm 2 The Process of Dynamic Calibration

Input:
Estimated Finger Tip Position: $Fingertip_{pos} = (x_{tip}, y_{tip}, z_{tip})$
Projected Point on the Screen: $Proj_{pos} = (x_{proj}, y_{proj}, z_{proj})$
Ray Vector from Finger Tip to Screen: $\vec{Ray}_{hover}$
Whether the projection is within boundary: *isIn*
Output:
Δ that will apply to $HandRoot_{pos}$
Process:
We mark $Fingertip'_{pos}, \Delta'$ ... as the variables in the last frame
When physical touch has not occurred yet:
if *isIn* == TRUE **AND** $|\vec{Ray}_{hover}| < Threshold$ **then**
  Calculate $\vec{v} = \frac{Fingertip_{pos} - Fingertip'_{pos}}{t - t'}$
  $\Delta = \Delta' - \vec{v} * (t - t') * \alpha$, where $\alpha \leq 1$
  Apply the new Δ to $HandRoot_{pos}$
end if
When physical touch occurs:
Get **2D** $TouchData = (a_{touch}, b_{touch}) \in PixelSpace$ from the touch device
Calculate **3D** position of Touch Position $Touch_{pos} = (x_{touch}, y_{touch}, z_{touch})$ in the virtual space, based on the Tracked phone data $\mathbf{M}_{PhoneToSpace}$ and $TouchData = (a_{touch}, b_{touch})$
Calculate **Spatial** difference $\delta = Touch_{pos} - Fingertip_{pos}$ and apply the δ onto the existing Δ' with proper method
Apply the new Δ to the $HandRoot_{pos}$ which represent the root joint of the whole hand

5 Study 2: Evaluating the Dynamic Calibration Algorithm

We conducted this second study to evaluate our Dynamic Calibration approach. The main objective was to assess target acquiring interactions with and without applying our technique. Based on previous research [40], we decided to use Oculus Quest 2 built-in hand tracking. We chose this hand tracking technique because of its widespread use. Furthermore, its accuracy is not as good as expensive external market-based solutions like Vicon, making it a good candidate for testing our dynamic calibration algorithm.

In this study, participants relied on visual feedback for the targeting process because different hand visual representations can influence the results of the task performance [23, 34, 43]. We also tested two variations of the virtual hand representations: Opaque and Transparent Hands (Figure 6).

5.1 Study Design

The major independent variables of this study are whether we applied the dynamic calibration algorithm or not (Marked as *Applied / NotApplied*), and hand representation (Marked as *Opaque / Transparent*). We employed two different types of *Distance* between targets(small: 24.7 mm, large: 60.3 mm) and two different types of target *Size*(diameter,small: 2.4mm, large: 7.2 mm) were applied.

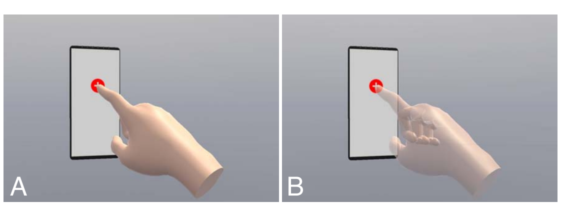

Figure 6: Evaluation conditions for virtual hand representations in Study 2: A) Opaque, and B) Transparent.

During the pilot, we found that the Oculus Quest 2 may lose the user's hand tracking or return unreliable tracking data if the participant holds and operates the phone with the same hand (like the Thumb condition in Study 1). So for the posture, we decided to use the same setup from Study 1's Index condition. Participants hold the phone with their non-dominant hand and operate with the dominant's hand index finger. Since we fixed the phone for the study, after the initial calibration, we considered the physical phone and its virtual representation to be the same, allowing us to focus more on the difference between virtual and physical hands during the study.

For *Accuracy*, we recorded *Z-Depth*, *Virtual Error Distance*, *Projected Error Distance* and *Physical Error Distance* for analysis. The *Z-Depth* refers to the perpendicular distance between the estimated fingertip to the phone plane; *Virtual Error Distance* is the distance between the estimated fingertip to the target; *Projected Error Distance* represents the distance between the projection point on the phone surface to the target; and the *Physical Error Distance* is the distance between physical touch to the physical target. The Estimated Finger Tip and the Projected Point are calculated based on the algorithm proposed in Section 4, whereas the physical touch point is retrieved from the phone directly. All distances are measured as Euclidean distances and recorded in *mm*. When physical contact occurs with the phone, we collected all types of distances based on the data at the frame before applying the Dynamic Calibration Algorithm. We measured the speed of target acquisition as *Time* (the elapsed time between a participant being presented with a target and their finger touching the screen).

In this study, we considered both on-screen-start and off-screen-start target acquisition tasks. Similarly to study 1, we instructed the participants to click on a mouse within their arm's length range but apart from the phone to achieve the off-screen-start target acquisition condition. In the on-screen-start target acquisition condition, the participants continuously acquired targets using the same finger and remained in the nearby area of the screen. Once the participants finished the on-screen trails, they would see a notification to start an off-screen operation on the virtual screen. The notification text will return to the target point once they click the mouse.

Inspired by previous works [8, 9, 47], we use a Balanced Latin Square to counter-balance Algorithm X Visualization (2 X 2), and we randomized the combinations of Size X Distance (2 X 2) within each significant condition. We generated four blocks containing nine trials as a group for each condition. The nine trials contained one Off-screen-start trial and eight On-screen-start trials. This approach means the design is a within-subjects study that two algorithms x 2 visualization x 2 sizes X 2 Distances x 4 blocks x 9 trials (1 off-screen + 8 on-screen) = 576 trials of touching per participant.

5.2 Apparatus and Set-Ups

We used a Google Pixel 6 Pro smartphone (6.7 inches, 1440 x 3120 pixels display with 512 pixels per inch) and a Oculus Quest 2 as our study platform.

From Study 1, we found that users tend to move their phone in the Index condition after several minutes. Due to the impact this could have on the overall performance, we decided to use a tripod to maintain the phone fixed. Participants were still instructed to hold the phone with their non-dominant hand, to maintain a body-relative frame that could influence the proprioception they have.

We developed the evaluation prototype in Unity 2020 LTS on a PC with a GTX 1080 6GB, and an Intel(R) Xeon(R) CPU E5-2630 with 64 GB as physical memory. The phone was connected to the PC using a customized WiFi 6 network to transfer the study's touch data via UDP.

5.3 Participants

We recruited 12 participants (five females) between the ages of 20 and 30 (Mean = 24.3, SD = 3.0). Three had reported having experience with VR/AR equipment, with one reporting a weekly usage exceeding 10 hours. All participants were right-handed and had normal or corrected-to-normal vision. The participant's education level varied from undergraduate to graduate. We offered compensation.

5.4 Procedure

We instructed the participants to adjust their body position until they were comfortable with the headset-to-phone distance. Similar to Study 1, we utilized visual widgets whenever the headset-to-phone distance changed significantly (20% of the head-phone distance compared to the initial), reminding participants to maintain the original preset posture. At the beginning of the study, we guided the participants to use their rendered opaque index fingers to touch the phone with the dynamic calibration algorithm. We employed a training session with randomized Size-Distance targets to familiarize users with the process. We also used that data to initialize the calibration and then conducted the formal study once the participants were confident with the task.

Each block started with a "Please put your finger back." instruction displayed on the virtual touchscreen for the Off-screen-start target acquisition task. Immediately after touch events, we displayed a new target. Then, we instructed the participants to perform On-screen-start target acquisition until they achieved all targets for each group. We instructed the participants to try acquiring targets quickly and accurately in all trials. Also, we showed subsequent targets even if participants missed that target. Participants were allowed to take a break between Algorithm X Visualization conditions. During their break, we would also ask the participants whether they have noticed anything strange to see if our algorithm was noticeable or not.

5.5 Results

We recorded data in both JSON and CSV format. We used *statsmodels* and *pandas* in python to process and analyze data. We applied a Repeated-measures analysis of variance (ANOVA-RM) for both On-Screen and Off-Screen data.

5.5.1 On-screen

From the ANOVA-RM we found that our algorithm provides significant effects on both z-depth ($F(1,11) = 132.319, p < .001$), Virtual Error Distance($F(1,11) = 104.977, p < .001$) and Physical Error Distance($F(1,11) = 9.314, p < .05$). The mean values on Z-Depth are 7.97 $\pm$ 0.63 mm to 16.53 $\pm$ 0.93mm regarding Applied to Not Applied. The mean values on Virtual Error Distance are 9.82 $\pm$ 0.80 mm to 17.51 $\pm$ 0.96mm regarding Applied to Not Applied. Also, the mean values on Physical Error Distance are 4.24 $\pm$ 0.25 mm to 5.75 $\pm$ 0.48mm regarding Applied to Not Applied.

Figure 7 shows the accuracy results between applying and not applying the Dynamic Calibration. We found that for these types of errors, our *Dynamic Calibration Algorithm* reduced errors significantly (52% for Z-Depth, 43% for Virtual Error Distance, 26% for Physical Error Distance). These results suggest that our approach can create more accurate touch performance both in the virtual and real world.

The ANOVA-RM did not reveal significant effects regarding different visualization methods on the four measured error distances. This might indicate that the different visualization methods (at least the Transparency) should not influence the performance. We did not find any significant effects regarding the Algorithm on the Projected Error Distance as well, using an ANOVA-RM. This is reasonable since this error distance refers to the user's intention of aiming at a target in VR, which our Dynamic Calibration Algorithm should not influence. The mean of Projected Error is 4.80 $\pm$ 0.43 mm, which is a relatively accurate result and can be used to evaluate our Finger Tip Estimating Algorithm. However, using a ANOVA-RM, we found significant effects regarding Target Distance on Physical Error Distance ($F(1,11) = 13.291, p < .005$). The mean values on

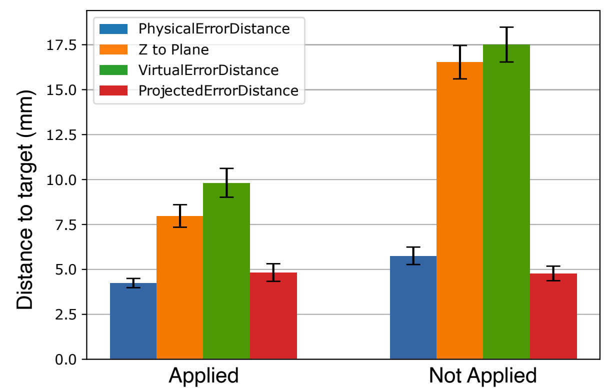

Figure 7: Touch Errors with and without Dynamic Calibration Algorithm

Physical Error Distance are 5.33 $\pm$ 0.33mm to 4.67 $\pm$ 0.30mm regarding Large to Small Target Distance. The ANOVA-RM also found significant effects regarding Target Distance on Projected Error Distance ($F(1,11) = 24.798, p < .001$). The mean values on Projected Error Distance are 5.14 $\pm$ 0.45mm to 4.45 $\pm$ 0.41mm regarding Large to Small Target Distance. Furthermore, using a ANOVA-RM, we found significant effects regarding Target Size on Projected Error Distance ($F(1,11) = 6.27, p < .05$). The mean values on Projected Error Distance are 4.64 $\pm$ 0.44mm to 4.96 $\pm$ 0.43mm from Small Size to Large. Yet, no significant effects regarding Target Distance were found on Virtual Error Distance and Z-Depth. And no Significant effects regarding Target Size were found on Virtual Error Distance, Physical Error Distance and Z-Depth. Related to the Speed, the ANOVA-RM reveal significant effects regarding Target Distance($F(1,11) = 146.792, p < .001$) and Target Size($F(1,11) = 19.865, p < .001$) on Time. The mean time of Large and Small Target Distance are 0.85 $\pm$ 0.05s and 0.70 $\pm$ 0.05s. And the mean time of Large and Small Target Size are 0.74 $\pm$ 0.05s and 0.81 $\pm$ 0.05s. Finally, we found no significant effects regarding Algorithm and Visualization on Time.

5.5.2 Off-screen

For the Off-Screen conditions, we found significant effects, using a ANOVA-RM, regarding Apply/Not Apply Dynamic Calibration Algorithm on Z-Depth ($F(1,11) = 84.457, p < .001$). The Applied mean = 7.43 $\pm$ 0.62mm and Not Applied mean = 16.47 $\pm$ 1.01mm) Also, the ANOVA-RM revel significant effects regarding Apply/Not Apply Dynamic Calibration Algorithm on Virtual Error Distance ($F(1,11) = 71.380, p < .001$), with Applied mean = 9.06 $\pm$ 0.69mm and Not Applied mean = 17.41 $\pm$ 1.03mm.

5.6 Discussion

We are encouraged to find out that our Dynamic Calibration Algorithm could significantly reduce both the Z-Depth Error and Virtual Touch Error Distance in On-Screen and Off-Screen conditions. Furthermore, our method reduced the Physical Touch Error Distance in On-Screen conditions. These results show that our algorithm could improve the overall performance of target acquisition for using a touch device in Virtual Reality.

Previous research has demonstrated that we can accept X-Y offsets if the virtual contact point determines touch positions [31,32,35, 40,41]. However, we cannot overlook the Z-Depth Errors since they might cause inconsistencies between visual and haptic. While we provided the statistically significant results based on ANOVA-RM, here we give the plots of all our participants' Trial to Z-depth data, as Figure 8 shows.

The Projected Distance Error results showed the performance of our fingertip estimation. The non-significance regarding the algorithm on the Projected Distance Error means that our algorithm does not influence how users aim before the touch. The significant effects of Target Distance and Target Size on the Projected Distance Error mean the user's target acquisition performance based on our Finger Tip Estimation follows Fitts' Law, strengthening our confidence about the estimation method in practice. The non-significant results regarding the visualization of any dependent variable are something we did not expect. Further investigation might be necessary according to this result.

In addition to the quantitative analysis, we asked participants for feedback after each condition. We specifically care about the feeling of "visual jump" caused by our algorithm, as this may be noticeable and influence the interaction performance. Ten out of 12 participants reported that they sometimes felt their finger being "moved". We will discuss this artifact of our technique in the following section.

6 Limitations and Future Work

In general, this research achieved our goals of proving both the proprioception and our Dynamic Calibration Algorithm can improve touch-device interactions in VR.

In Study 1, we only studied three different postures to represent different" body-spatial relative frame" levels. Future work should investigate more posture conditions [29] that combines further hand usage (i.e., hold the phone with two hands), phone orientation (portrait or landscape), and operation conditions (i.e., standing and walking) to figure out more concrete results in proprioception studies. In Study 2, our phone was in a fixed position after a one-time calibration. As we mentioned in the study design, the purpose is to prevent the user from moving the phone during the study, a trend we found in Study 1 of the INDEX condition. Though it is common to use a fixed touch surface for evaluating touch performance in Spatial Environment [40,51], this adaption might weaken the "body-relative frame" and thus influence the participants' proprioception.

More than proprioception, if we allowed the phone to move while tracking, the tracking errors of the phone should also be considered. Our dynamic calibration algorithm should still work since we fix the relative errors rather than the absolute ones. However, we cannot predict what context our algorithm should majorly rely on, as it could either calibrate the finger towards the phone, reversely calibrate the smartphone towards the finger, or use a combination of them. Future work should focus on answering this question. Though we claimed that our dynamic calibration methods could be universal regardless of the tracking techniques for hands, we only focused on evaluating the algorithm in Oculus Quest 2 first. Other tracking techniques should be studied in the future, including to Vive Built-in Camera [40,41,43], Leapmotion [40,41,45], segmented video pass through [27,34]. We could also do those evaluations with an external system like Vicon as a baseline to see if our algorithm could also improve the absolute tracking accuracy instead of relative offset.

We limited our participants' interactions to the same finger for both studies. While this is reasonable for a touch-based study, previous works [13,20,43] have shown the influence of different fingers on both performance and users' willing. Our dynamic calibration did not accomplish primary finger estimation and assumed the primary finger in advance. So applying the finger estimation that allows users to change their touch finger could be deployed in the future.

Furthermore, our task focused on target acquisition with relatively small targets (within finger's size), a scenario distant from users' everyday UI. Other tasks (like typing and swiping) and applications closer to day-to-day life should be further evaluated.

For both studies, the total number of participants is relatively small, with a relatively shallow range of ages, which may influence the results. Studies with more participants and a more comprehensive range of ages might be a good option for future research.

In our second study, we only applied two types of visualization techniques (transparent and opaque). Other visualization methods, like different sizes of the hand and fingers [23,50], video see-through fingers [5,34] and different avatars like animal-style hand [28] could be subject of future studies.

As reported in section 5.6, participants were sometimes aware of the effects of visual jump in the hand representation when a relatively big Δ was applied in our Dynamic Calibration Algorithm. This effect might occur due to our algorithm not applying a smooth transition when calibrating results. Yet we don't know whether the "sudden jump" influences the touch performance or not. Future research should study a smooth transition to see if it can help users perform more naturally.

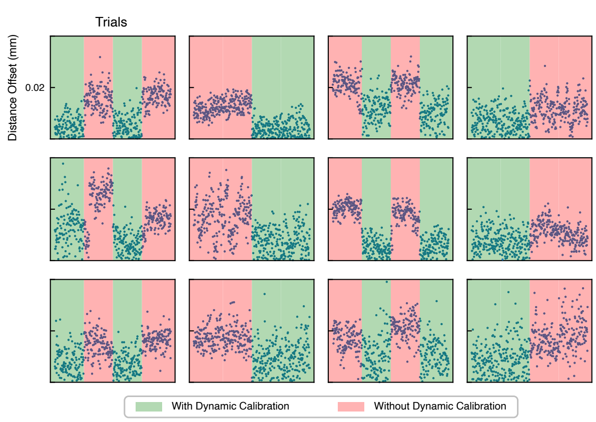

Figure 8: Participants' Z-Depth distance offset data per trial with and without applying the Dynamic Calibration Algorithm.

7 Conclusions

Touch-based interactions using smartphones or tablet devices can be a reliable and accurate input modality in VR immersive experiences. Yet, interacting with touch-enabled devices can be troublesome in commercial HMDs due to the usually employed inside-out tracking solutions. Despite being adequate for most usage scenarios, these technologies are not accurate enough for people to perform precise touch gestures on physical devices while immersed in VR. In this paper, we tackle the issue of how to reduce the miss-alignment between physical hands and touch devices with their virtual representations that can impair touch interactions and break the realism. Therefore, we presented a proprioception user study to evaluate touch performance without providing a rendering of the participants' hands in VR. This first study suggests that the posture used to hold touch devices and the distance to the target can influence accuracy. We then contributed a dynamic calibration method to reduce the miss-alignment between the user's physical hand and its virtual representation and thus, improve the precision of touch. We evaluated our dynamic calibration algorithm in a final user study. The results suggest that our approach can improve touch accuracy and helps reduce common depth-based virtual hand representation errors. Finally, we hope our dynamic calibration approach will improve touch-based interactions in VR and inform future immersive touch-based user interfaces.

Acknowledgments

The authors would like to thank the anonymous reviewers. We also would like to thank our colleague Jiannan Li for all the help he was kind to provide. This research was supported in part by the National Sciences and Engineering Research Council of Canada (NSERC) under Grant IRCPJ 545100 - 18.

REFERENCES

[1] M. Al-Sada, F. Ishizawa, J. Tsurukawa, and T. Nakajima. Input forager: A user-driven interaction adaptation approach for head worn displays. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia*, pp. 115–122, 2016.

[2] R. Arora, R. Habib Kazi, T. Grossman, G. Fitzmaurice, and K. Singh. Symbiosissketch: Combining 2d & 3d sketching for designing detailed 3d objects in situ. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, p. 1–15. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3173574.3173759

[3] R. Arora, R. H. Kazi, F. Anderson, T. Grossman, K. Singh, and G. Fitzmaurice. Experimental evaluation of sketching on surfaces in vr. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, p. 5643–5654. Association for Computing Machinery, New York, NY, USA, 2017. doi: 10.1145/3025453.3025474

[4] M. Azmandian, M. Hancock, H. Benko, E. Ofek, and A. D. Wilson. Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. In *Proceedings of the 2016 chi conference on human factors in computing systems*, pp. 1968–1979, 2016.

[5] H. Bai, L. Zhang, J. Yang, and M. Billinghurst. Bringing full-featured mobile phone interaction into virtual reality. *Computers & Graphics*, 97:42–53, 2021.

[6] A. U. Batmaz, A. K. Mutasim, M. Malekmakan, E. Sadr, and W. Stuerzlinger. Touch the wall: Comparison of virtual and augmented reality with conventional 2d screen eye-hand coordination training systems. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 184–193. IEEE, 2020.

[7] P. Baudisch and G. Chu. Back-of-device interaction allows creating very small touch devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, p. 1923–1932. Association for Computing Machinery, New York, NY, USA, 2009. doi: 10.1145/1518701.1518995

[8] X. Bi, Y. Li, and S. Zhai. Ffitts law: Modeling finger touch with fitts' law. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, p. 1363–1372. Association for Computing Machinery, New York, NY, USA, 2013. doi: 10.1145/2470654.2466180

[9] X. Bi and S. Zhai. Predicting finger-touch accuracy based on the dual gaussian distribution model. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, p. 313–319. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2984511.2984546

[10] V. Biener, D. Schneider, T. Gesslein, A. Otte, B. Kuth, P. O. Kristensson, E. Ofek, M. Pahud, and J. Grubert. Breaking the screen: Interaction across touchscreen boundaries in virtual reality for mobile knowledge workers. *IEEE Transactions on Visualization and Computer Graphics*, 26(12):3490–3502, 2020.

[11] T. Carter, S. A. Seah, B. Long, B. Drinkwater, and S. Subramanian. Ultrahaptics: Multi-point mid-air haptic feedback for touch surfaces. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, p. 505–514. Association for Computing Machinery, New York, NY, USA, 2013. doi: 10.1145/2501988.2502018

[12] L.-W. Chan, H.-S. Kao, M. Y. Chen, M.-S. Lee, J. Hsu, and Y.-P. Hung. Touching the void: Direct-touch interaction for intangible displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, p. 2625–2634. Association for Computing Machinery, New York, NY, USA, 2010. doi: 10.1145/1753326.1753725

[13] A. Colley and J. Häkkilä. Exploring finger specific touch screen interaction for mobile phone user interfaces. In *Proceedings of the 26th Australian computer-human interaction conference on designing futures: The future of design*, pp. 539–548, 2014.

[14] T. Dorta, G. Kinayoglu, and M. Hoffmann. Hyve-3d and the 3d cursor: Architectural co-design with freedom in virtual reality. *International Journal of Architectural Computing*, 14(2):87–102, 2016.

[15] D. Elliott, J. Lyons, and K. Dyson. Rescaling an acquired discrete aiming movement: Specific or general motor learning? *Human Movement Science*, 16(1):81–96, 1997.

[16] C. M. Fang and C. Harrison. Retargeted self-haptics for increased immersion in vr without instrumentation. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pp. 1109–1121, 2021.

[17] R. Fares, S. Fang, and O. Komogortsev. Can we beat the mouse with magic? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, p. 1387–1390. Association for Computing Machinery, New York, NY, USA, 2013. doi: 10.1145/2470654.2466183

[18] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381, 1954.

[19] M. Goel, J. Wobbrock, and S. Patel. Gripsense: using built-in sensors to detect hand posture and pressure on commodity mobile phones. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*, pp. 545–554, 2012.

[20] A. Goguey, M. Nancel, G. Casiez, and D. Vogel. The performance and preference of different fingers and chords for pointing, dragging, and object transformation. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 4250–4261, 2016.

[21] E. J. Gonzalez, P. Abtahi, and S. Follmer. Reach+ extending the reachability of encountered-type haptics devices through dynamic redirection in vr. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pp. 236–248, 2020.

[22] J. Grubert, M. Heinisch, A. Quigley, and D. Schmalstieg. Multifi: Multi fidelity interaction with displays on and around the body. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 3933–3942, 2015.

[23] J. Grubert, L. Witzani, E. Ofek, M. Pahud, M. Kranz, and P. O. Kristensson. Effects of hand representations for typing in virtual reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 151–158. IEEE, 2018.

[24] K. Hasan, X.-D. Yang, H.-N. Liang, and P. Irani. How to position the cursor? an exploration of absolute and relative cursor positioning for back-of-device input. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '12, p. 103–112. Association for Computing Machinery, New York, NY, USA, 2012. doi: 10.1145/2371574.2371591

[25] C. Holz and P. Baudisch. Understanding touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, p. 2501–2510. Association for Computing Machinery, New York, NY, USA, 2011. doi: 10.1145/1978942.1979308

[26] H. Kim, M. Kim, and W. Lee. Hapthimble: A wearable haptic device towards usable virtual touch screen. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, p. 3694–3705. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2858036.2858196

[27] W. Kim, K. T. W. Choo, Y. Lee, A. Misra, and R. K. Balan. Empath-d: Vr-based empathetic app design for accessibility. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 123–135, 2018.

[28] A. Krekhov, S. Cmentowski, and J. Krüger. The illusion of animal body ownership and its potential for virtual reality games. In *2019 IEEE Conference on Games (CoG)*, pp. 1–8. IEEE, 2019.

[29] S. Lee, G. Kyung, J. Lee, S. K. Moon, and K. J. Park. Grasp and index finger reach zone during one-handed smartphone rear interaction: effects of task type, phone width and hand length. *Ergonomics*, 59(11):1462–1472, 2016.

[30] R. W. Lindeman, J. L. Sibert, and J. K. Hahn. Hand-held windows: towards effective 2d interaction in immersive virtual environments. In *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*, pp. 205–212. IEEE, 1999. doi: 10.1109/VR.1999.756952

[31] P. Lubos, G. Bruder, O. Ariza, and F. Steinicke. Touching the sphere: Leveraging joint-centered kinespheres for spatial user interaction. In *Proceedings of the 2016 Symposium on Spatial User Interaction*, SUI '16, p. 13–22. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2983310.2985753

[32] P. Lubos, G. Bruder, and F. Steinicke. Analysis of direct selection in

head-mounted display environments. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 11–18. IEEE, 2014. doi: 10.1109/3DUI.2014.6798834

[33] A. Makhsadov, D. Degraen, A. Zenner, F. Kosmalla, K. Mushkina, and A. Krüger. Vrysmart: a framework for embedding smart devices in virtual reality. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pp. 1–8, 2022.

[34] F. Matulic, A. Ganeshan, H. Fujiwara, and D. Vogel. 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*, CHI '21. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3411764.3445583

[35] T. Menzner, T. Gesslein, A. Otte, and J. Grubert. Above surface interaction for multiscale navigation in mobile virtual reality. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 372–381, 2020. doi: 10.1109/VR46266.2020.00057

[36] A. Millette and M. J. McGuffin. Dualcad: integrating augmented reality with a desktop gui and smartphone interaction. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, pp. 21–26. IEEE, 2016.

[37] M. R. Mine, F. P. Brooks, and C. H. Sequin. Moving objects in space: Exploiting proprioception in virtual-environment interaction. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, p. 19–26. ACM Press/Addison-Wesley Publishing Co., USA, 1997. doi: 10.1145/258734.258747

[38] Y. Monnai, K. Hasegawa, M. Fujiwara, K. Yoshino, S. Inoue, and H. Shinoda. Haptomime: Mid-air haptic interaction with a floating virtual screen. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST '14, p. 663–667. Association for Computing Machinery, New York, NY, USA, 2014. doi: 10.1145/2642918.2647407

[39] E. Normand and M. J. McGuffin. Enlarging a smartphone with ar to create a handheld vesad (virtually extended screen-aligned display). In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 123–133. IEEE, 2018.

[40] D. Schneider, V. Biener, A. Otte, T. Gesslein, P. Gagel, C. Campos, K. Čopič Pucihar, M. Kljun, E. Ofek, and M. Pahud. Accuracy Evaluation of Touch Tasks in Commodity Virtual and Augmented Reality Head-Mounted Displays. In *Symposium on Spatial User Interaction*, pp. 1–11, 2021.

[41] D. Schneider, A. Otte, A. S. Kublin, A. Martschenko, P. O. Kristensson, E. Ofek, M. Pahud, and J. Grubert. Accuracy of Commodity Finger Tracking Systems for Virtual Reality Head-Mounted Displays. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pp. 804–805, Mar. 2020. doi: 10.1109/VRW50115.2020.00253

[42] K. A. Siek, Y. Rogers, and K. H. Connelly. Fat finger worries: how older and younger users physically interact with pdas. In *IFIP Conference on Human-Computer Interaction*, pp. 267–280. Springer, 2005. doi: 10.1007/11555261_24

[43] J. Son, S. Ahn, S. Kim, and G. Lee. Effect of contact points feedback on two-thumb touch typing in virtual reality. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, CHI EA '22. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10.1145/3491101.3519794

[44] H. B. Surale, A. Gupta, M. Hancock, and D. Vogel. Tabletinvr: Exploring the design space for using a multi-touch tablet in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–13. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290605.3300243

[45] P. P. Valentini and E. Pezzuti. Accuracy in fingertip tracking using leap motion controller for interactive virtual applications. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 11(3):641–650, 2017.

[46] D. Valkov, A. Giesler, and K. Hinrichs. Evaluation of depth perception for touch interaction with stereoscopic rendered objects. In *Proceedings of the 2012 ACM international conference on Interactive tabletops and surfaces*, pp. 21–30, 2012.

[47] D. Vogel and P. Baudisch. Shift: a technique for operating pen-based interfaces using touch. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 657–666, 2007.

[48] D. Wigdor, C. Forlines, P. Baudisch, J. Barnwell, and C. Shen. Lucid touch: A see-through mobile device. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, p. 269–278. Association for Computing Machinery, New York, NY, USA, 2007. doi: 10.1145/1294211.1294259

[49] K. Wolf, C. Müller-Tomfelde, K. Cheng, and I. Wechsung. Does proprioception guide back-of-device pointing as well as vision? In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, p. 1739–1744. Association for Computing Machinery, New York, NY, USA, 2012. doi: 10.1145/2212776.2223702

[50] H. Xia, T. Grossman, and G. Fitzmaurice. Nanostylus: Enhancing input on ultra-small displays with a finger-mounted stylus. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, p. 447–456. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2807442.2807500

[51] R. Xiao, J. Schwarz, N. Throm, A. D. Wilson, and H. Benko. Mrtouch: Adding touch input to head-mounted mixed reality. *IEEE transactions on visualization and computer graphics*, 24(4):1653–1660, 2018.

[52] F. Zhu and T. Grossman. *BISHARE: Exploring Bidirectional Interactions Between Smartphones and Head-Mounted Augmented Reality*, p. 1–14. Association for Computing Machinery, New York, NY, USA, 2020.