Stanislav S. Kyian\* Carleton University, Canada stanislavkyian@cmail.carleton.ca Robert J. Teather Carleton University, Canada rob.teather@carleton.ca

## ABSTRACT

We present a method to track a smartphone in VR using a fiducial marker displayed on the screen. Using WebRTC transmission protocol, we capture smartphone touchscreen input and the screen contents, copying them to a virtual representation in VR. We present two Fitts' law experiments assessing the performance of selecting targets displayed on the virtual smartphone screen using this method. The first compares direct vs. indirect input (i.e., virtual smartphone co-located with the physical smartphone, or not), and reveals there is no difference in performance due to input indirection. The second experiment assesses the influence of input scaling, i.e., decoupling the virtual cursor from the actual finger position on the smartphone screen so as to provide a larger virtual tactile surface. Results indicate a small effect for extreme scale factors. We discuss implications for the use of smartphones as input devices in VR.

#### CCS CONCEPTS

• Human-centered computing → Human computer interaction (HCI); Interaction devices; Touch screens; Human computer interaction (HCI); Interaction techniques; Pointing; Human computer interaction (HCI); Interaction paradigms; Virtual reality.

#### **KEYWORDS**

Virtual Reality, Smartphone, Touchscreen, Selection task

#### **ACM Reference Format:**

Stanislav S. Kyian<sup>\*</sup> and Robert J. Teather. 2021. Selection Performance Using a Smartphone in VR with Redirected Input. In *Symposium on Spatial User Interaction (SUI '21), November 09, 10, 2021, Virtual Event, USA*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3485279.3485292

#### **1** INTRODUCTION

Many VR head-mounted displays (HMDs) now provide full 6 degrees-of-freedom (6DOF) pose tracking out of the box via tracked controllers with several buttons and joysticks in addition to motion tracking capabilities. Such controllers are not universally available, e.g., on smartphone-based VR devices such as Google Cardboard. Smartphones offer a potential opportunity as an interaction device

SUI '21, November 09, 10, 2021, Virtual Event, USA

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-9091-0/21/11...\$15.00

https://doi.org/10.1145/3485279.3485292

in VR. In recent years, smartphones have acquired many new capabilities, becoming full-fledged personal computers with various sensors such as gyroscopes that provide good orientation tracking and accelerometers, cameras, and GPS sensors for position tracking. These features are used in much cheaper smartphone-based VR HMD options, such as Google Cardboard<sup>1</sup>. However, this is just one example of VR-smartphone integration; smartphones offer potential not just as a display, but also as VR interaction devices.

Given the ubiquity of smartphones, it would be beneficial for VR users to be able to use a smartphone in VR, without context switching. Consider playing a VR game: if the player receives a call or message on their smartphone, they must remove the VR headset, use the smartphone, and put the headset back again when done. This impacts immersion and causes breaks in presence. Moreover, the smartphone touchscreen provides a potentially useful interaction surface, offering the well-known benefits of passive haptic feedback in VR [20, 24, 26]. Recent VR headsets, such as the Oculus Quest are equipped with outward-facing cameras. These cameras can support optical smartphone tracking, enabling direct use of a smartphone in VR either to avoid breaks in presence (i.e., "general purpose" smartphone usage in VR) or as an input device. Supporting smartphone tracking in VR may increase the use of HMDs in users' routines, by streamlining smartphone usage. This could be used in VR teleconferencing (e.g., managing a presentation). Even low-cost smartphone-based HMDs (e.g., Google Cardboard) can use the outward-facing smartphone camera to support tracking a second smartphone. This may be especially useful in providing improved interaction capabilities for budget HMDs that otherwise do not include a 3D tracked controller (assuming a user has two smartphones available).

There are several studies examining the use of smartphones or wearable devices as VR input devices [2, 6, 15, 18, 24, 46–27, 36, 44]. To date, none tracked the smartphone's 6DOF pose without attaching a bulky VR controller or tracking device to it. While affixing a VR controller to a smartphone provides good tracking, it requires an extra VR controller or sensor and makes the smartphone less ergonomic. Thus, the smartphone becomes uncomfortable and awkward to use both in and out of VR until detaching the controller from the smartphone. Tracking systems like Vicon<sup>2</sup>, are prohibitively expensive and generally constrained to one room. These limitations make it impractical to track a smartphone in VR in such a fashion. Moreover, with the exception of recent work by Bai et al. [4], none of the previous solutions supported integrating the smartphone's capabilities in VR, to support normal smartphone usage.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

<sup>&</sup>lt;sup>1</sup>https://arvr.google.com/cardboard/

<sup>&</sup>lt;sup>2</sup>https://www.vicon.com/

Currently, there are virtually no mobile apps that allow a smartphone to be used in VR without additional hardware, and few studies on using a smartphone as a VR controller. We propose a technique using an on-screen display of fiducial markers to support a 6DOF tracking of a smartphone in VR, requiring no additional sensors. Our method also presents a virtual replica of the smartphone in VR and copies the smartphone screen to the virtual replica. The user can thus interact with the smartphone in VR similar to reality.

Our paper presents two main contributions. First, is our method of tracking the smartphone itself, and using it as a tactile surface in VR. In our case, we used a Google Cardboard-like smartphone VR device, but our approach also applies to other HMDs with outwardfacing cameras. Second, we present two experiments on the effectiveness of using a smartphone in VR. The VR smartphone need not perfectly replicate realistic input, introducing the possibility of redirected input. We thus investigate co-location and input scaling to determine which offers the optimal input conditions VR smartphone use.

### 2 RELATED WORK

# 2.1 Smartphone usage in VR

Several projects have looked at incorporating smartphones into VR. While modern smartphones offer several features useful for VR, interaction on planar surfaces (e.g., a touchscreen) is also beneficial in VR selection and manipulation [20, 24, 26]. Amano et al. [2] used VR as a testbed for smartphone navigation maps using GPS and Wi-Fi to locate the user in the map. The system captured the smartphone screen and streamed it to a PC over USB, which then presented it in VR. The authors reported latency on the order of 695.5ms to 729.0ms, which is over double what we achieved using the WebRTC protocol. Such high latency is impractical for interactive applications. Huidong et al. [4] attached a VR controller to a smartphone for pose tracking and attached a depth camera to the HMD to capture hand gestures. The smartphone display was mirrored in real-time in VR, although the authors did not specify how. They reported using a private network connection with a 170 ms average delay. The authors report that user preference questionnaires for common smartphone tasks (e.g., calling, taking pictures, using social media apps) were significantly better for reality. Of the VR conditions, calling and photography had the highest subjective preference scores while typing instant messages had the lowest. Yet, their application required depth cameras which to date, are not readily available on VR devices.

Other studies used a smartphone in VR as a touch input device or VR controller. Mohr et al. [33] introduced TrackCap, which used the Vuforia SDK<sup>3</sup> to track a fiducial marker mounted a Microsoft Hololens<sup>4</sup>. Fusing the smartphone's 3DOF gyroscopic data with the smartphone's fiducial marker tracking allowed detection of the smartphone's 6DOF pose relative to the HMD. Experiments comparing selection and manipulation performance with TrackCap to gyroscope-only tracking revealed TrackCap offered significantly better performance. While the technique also switched between cameras depending on the smartphone orientation, the authors reported significant delay during camera switching and concluded this feature was impractical.

Gebhardt et al. [15] used a smartphone as a menu system input in a VR application presented in a VR CAVE, thus did not implement tracking or screencasting. An expert review and user study revealed that experts appreciated the system, but user acceptance was lower than expected. Menzner et al. [32] compared an above surface interaction technique with a 2D on-surface input for VR map navigation. The system used a smartphone as a surface and tracked both the smartphone and the user's fingers with an OptiTrack system. While above-surface interaction performed significantly better, the Opti-Track requires external tracking and markers.

# 2.2 Direct vs. Indirect Input

Since the user cannot actually see the physical smartphone in VR, neither the virtual smartphone nor its input space needs to be physically co-located with the physical smartphone or use a 1:1 input mapping. This introduces the possibility of having a larger virtual surface than that physically provided by the smartphone or positioning the virtual smartphone somewhere more comfortable. Results on direct vs. indirect input have yielded mixed results. For example, Forlines et al. [11] found that users benefit from direct-touch input on bimanual tasks. Another study [12] on pointing and crossing tasks found that while direct input was significantly more efficient in crossing, direct and indirect were equivalent in point selection. Other research indicated that older adults were more influenced by the input device choice [31].

Other research [39] on 3D target selection did not find significant differences between co-located and disjoint input spaces. However, Imamov et al. [19] found that context switching time increases if the information was displayed far from the task position. Participants found it more preferable when content was placed at eye level or below and positioned at medium distances.

Introducing mismatches between the physical and virtual screen plane positions and employing surface warping are examples of input redirection. Haptic retargeting, such as Azmandian's approach. [3], rotates the immersive virtual environment to align real and virtual objects [45]. Azmandian et al. [3] conducted a study where such retargeting was applied to create the illusion of multiple tangible cubes in the VR environment, despite only using a single physical. They found the highest presence and satisfaction scores when combining body and world warping.

Other research [16] compared two remapped reaching techniques: static translational offset between the virtual and physical hand before a reaching action and one that dynamically interpolates the position of the virtual hand during a reaching motion. This is similar to the well-known Go-Go technique [35]. The results showed that the static translational shift performed better than the dynamic one and was more robust for situations with larger mismatches between virtual and physical objects.

Overall, results generally show little difference between direct and indirect conditions, as suggested by the prism adaptation research [1, 5, 8, 13, 14]. This is promising for designing smartphonebased interaction in VR, as it supports decoupling the physical and virtual smartphone positions for ergonomic reasons.

<sup>&</sup>lt;sup>3</sup>https://developer.vuforia.com/

<sup>&</sup>lt;sup>4</sup>https://www.microsoft.com/hololens

SUI '21, November 09, 10, 2021, Virtual Event, USA

# 2.3 Effects of Scale on Selection Performance

Another type of input scaling is control-display (CD) gain, which is not typically used with touchscreen input. CD gain can be applied in VR to increase the size of an interaction surface. This is potentially useful to make small tactile surfaces (e.g., smartphones, smart-watches) behave like a larger virtual surface. Among various possible applications, CD gain can increase the smartphone thumb-zone without drastically decreasing efficiency.

There have been several studies on warped input surfaces in VR. Kohli et al. [22] used the Fitts' law reciprocal tapping task [10, 38] to compare one-to-one and scaled conditions on a plane oriented at various angles. Non-inferiority statistical testing indicated that selecting warped targets was statistically no worse than selection without scaling (1-to-1). A follow-up study [23] found that with training, participant performance was no worse in a warped virtual space than in an unwarped one. Didehkhorshid et al. [9] scaled stylus input on a tracked drawing tablet in VR, comparing selection performance across different scale factors (from 1-to-1 to a 2.4 scale factor). While scale factor significantly affected movement time, the throughput and error rate was no worse with scaling than the 1-to-1 mapping, similar to the results of Kohli et al. [22]. Thus, there is evidence that small scale factors have relatively little impact on performance. A goal of our study is to verify if such techniques can be applied with the smaller touchscreen surface of a smartphone.

# 2.4 Fitts' Law

We employ the Fitts' law selection task in our study, so briefly describe it here. Fitts' law predicts selection time as a function of target size and distance [34]. The model relies on these equations:

$$MT = a + b \cdot ID$$
 where  $ID = \log_2\left(\frac{A}{W} + 1\right)$  (1)

$$TP = \frac{ID_e}{MT} \quad where \quad ID_e = \log_2\left(\frac{A_e}{W_e} + 1\right)$$
(2)

$$A_e = \frac{\sum_{i=1}^N A_i}{N} \tag{3}$$

$$W_e = 4.133 \cdot SD_x \tag{4}$$

In Equation 1 *MT* is movement time, and *a* and *b* are empirically calculated via linear regression. The *ID* is the index of difficulty (the overall selection difficulty), based on the amplitude A – the distance the cursor travels to select the target, and *W*, the target width (i.e., target size). As seen in Equation 1, the index of difficulty will thus increase by increasing the amplitude and/or decreasing the target width.

Equation 2 presents throughput (*TP*), which combines speed and accuracy and is unaffected by the speed-accuracy trade-off commonly seen in such tasks [28].  $ID_e$  is the effective index of difficulty and gives difficulty of the task users actually performed, rather than the task presented. It is thus widely used and recommended by the ISO 9241-9 standard as a primary metric for pointing device comparison. In contrast, selection time and error rate are less consistent between studies, as they vary with individual participant biases towards speed or accuracy. Throughput has been shown to be more stable in both 2D and 3D selection studies [7, 37, 42, 43].



Figure 1: Fitts' law task, arrows indicate where the target would move after each selection.

Equation 3 presents effective amplitude,  $A_e$ , the mean movement distance between targets. Finally, Equation 4 presents effective width,  $W_e$ . This first requires calculating  $SD_x$ , the standard deviation of selection endpoints projected onto the vector between the two targets. It incorporates selection coordinate variability and is multiplied by 4.133, which corresponds to  $\pm 2.066$  standard deviations from the mean. This effectively resizes targets postexperimentally, such that 96% of selections hit the target. This process normalizes the experimental error rate to 4%. This accuracy adjustment facilitates the comparison of throughput scores between studies with varying error rates by first normalizing accuracy [29]. Figure 1 illustrates the selection task. Users must select the highlighted target repeatedly; the target moves as indicated by the arrows showing the target location for the next selection trial.

# **3 VR SMARTPHONE IMPLEMENTATION**

This section details and rationalizes the implementation of the prototype used in our studies. While the proposed method for presenting a smartphone in VR could be implemented using almost any HMD, we decided to first experiment with Cardboard VR<sup>5</sup> to facilitate remote experiments during the COVID-19 pandemic. Such devices typically have limited interaction capabilities, thus we were also interested if our technique improved smartphone-based VR interaction. Our current implementation thus required two smartphones: the SVR device (**S**martphone for **VR**, used in the HMD), and the SHI device (**S**martphone for **H**and Input, held by the user during the study. The system software consisted of three applications, VRPhone, FittsStudy, and RTC. All project source code is available on GitHub<sup>6</sup>. Descriptions of each application follow.

# 3.1 VRPhone

We developed VRPhone in Unity to run on the SVR device to present the virtual environment seen by the user, and track and display the second smartphone. VRPhone performs 6DOF tracking of the handheld SHI smartphone in real-time via a fiducial marker displayed on the SHI screen and displays corresponding virtual models using the SHI pose data. The fiducial marker (presented on the SHI device) is seen on the top half of the virtual smartphone. The application

<sup>&</sup>lt;sup>5</sup>https://developers.google.com/cardboard

<sup>&</sup>lt;sup>6</sup>https://github.com/Staskkk/VRPhone.git



Figure 2: VRPhone application screenshot during the selection task. (Left) Direct mode, where the virtual smartphone is collocated with the actual SHI device position. (Right) Indirect mode, where the virtual smartphone is in a fixed position.

also scans a QR-code displayed on the SHI device provided by the RTC application to establish a WebRTC connection between the two devices. See Figure 2

For camera-based tracking of the fiducial marker, we used the So-IAR framework, based on OpenCV<sup>7</sup> which provides robust markertracking performance for a smartphone. SoIAR sometimes loses marker tracking, which increases virtual smartphone model lag. To mitigate this, we also constantly send the SHI gyroscope data via the WebRTC UDP data channel to provide the virtual model orientation when the fiducial marker tracking is lost, and between rotation updates from the SoIAR framework. Gyroscope data is used relative to the last orientation received from the marker-tracking library. An absolute gyroscope orientation was not used because of the noticeable shift between SVR and SHI compass values and their measurement errors. Study-related data (condition, etc.) was sent via WebRTC over the TCP transport layer.

The software also presented the virtual model of the smartphone in either *direct* or *indirect* mode. In direct mode, the virtual smartphone screen was co-located with the virtual smartphone, much like a real smartphone. The direct mode requires 6DOF tracking of the SHI smartphone. In indirect mode, the virtual screen was rendered floating in space at a constant position above the virtual table, slightly below the user's eye level. The screen always rotated around its vertical axis to orient itself facing the user. The virtual smartphone model was still displayed but showed a black screen so as to keep the smartphone GPU and CPU load similar for both conditions. This feature is optional for indirect mode; both it and the camera tracking could be disabled in indirect mode, which would significantly decrease heat and battery usage of the smartphone.

For the "cardboard" display mode we used "Google VR SDK for Unity"<sup>8</sup> – a free SDK for VR Cardboard app development provided under MIT license. The virtual environment presented a small lab room with a table in front of the user and some furniture around the room. For the virtual smartphone model, we used a Samsung Galaxy S10 from TurboSquid<sup>9</sup> under academic usage allowance. For the QR-code scanning, we showed the smartphone's camera feed on the screen with an overlay hint at the start of the app. The scanning feature was provided by the ZXing library<sup>10</sup> (Apache-2.0 license). WebRTC protocol was employed from the MixedReality-WebRTC library<sup>11</sup> by Microsoft (MIT License).

Finally, since the user could not see their hands/fingers in VR, the SVR device also displayed a cursor on the virtual smartphone screen (in both direct and indirect modes) to show the contact point on the screen. This cursor was optionally subject to scaling, employing the warped virtual surfaces (WVS) technique described by Didehkhorshid et al. [9]. We used this technique in our second experiment. The method uses a linear proportion relative to an origin point, based on the specified scale factor specified (SF). For example, when the SF equals 1, the cursor moves in 1-to-1 mapping with the finger on the touchscreen. WVS allows users to interact with an arbitrary-sized virtual surface, while the physical interaction space or touchscreen is always the same (e.g., a touchscreen). We implemented the VWS technique to determine if it could provide easier access to the targets on the edges or outside of the thumb-zones of a touchscreen. Our study is also the first to test such a technique with one-handed finger input. Since we use only part of the touchscreen, the results may also indicate the applicability of VWS to small wearable devices (e.g., smartwatches) with a touchscreen.

## 3.2 FittsStudy

The FittsStudy application ran on the SHI device and was based on FittsLawUnity<sup>12</sup> by Hansen et al. [17] provided with a BSD-3-Clause license. The software presents a Fitts' law reciprocal selection task for different input devices such as a mouse, eye gaze, joystick, etc. We modified the code to add finger input, made adjustments to the logs, study configuration settings, and add Android OS support. The app sends the display condition parameters to the RTC app via the Android Broadcast method. We shifted the center of the target's circle lower to improve suitability for one-handed thumb-based input. To track the device during the study, we incorporated

<sup>&</sup>lt;sup>7</sup>https://opencv.org

<sup>&</sup>lt;sup>8</sup>https://github.com/googlevr/gvr-unity-sdk

<sup>9</sup>https://www.turbosquid.com

<sup>&</sup>lt;sup>10</sup>https://github.com/zxing/zxing

<sup>&</sup>lt;sup>11</sup>https://microsoft.github.io/MixedReality-WebRTC/index.html

<sup>12</sup>https://github.com/GazeIT-DTU/FittsLawUnity



Figure 3: Fitts application screenshots. (Left) Study configuration menu. (Right) Study during execution, depicting a fiducial marker at the top of the screen to support SHI device tracking. This also depicts the screen, captured and transmitted to the SVR device, as described above.

a fiducial marker image on the upper part of the screen. Finally, we implemented a script that emailed data logs directly to the experimenter at the end of the study. See Figure 3

## 3.3 RTC app

The RTC app also ran on the SHI device. It did not require graphics, so we developed it in Android Studio rather than Unity to make the project smaller and increase compilation speed. It was based on the ScreenShareRTC project<sup>13</sup> under the Apache-2.0 license. The RTC app copied the smartphone screen from the SHI device to the virtual smartphone displayed in the SVR application via a local Wi-Fi network at 1280×720 pixels resolution. This was sufficient to both clearly render the SVR screen and minimize latency.

During development, we found that the WebRTC protocol with Android screen-capturing greatly outperformed screenshots sent over UDP, the method used in another study [2]. WebRTC requires a signaling server to establish communication between endpoints. Therefore, we implemented a signaling server in the RTC app as a separate service and used the ZXing library (same as in VRPhone) to create a QR-code with the IP address of the smartphone. The SVR device then scanned the QR-code to establish the connection.

## 3.4 Latency Measurement

We measured screen transmission latency by recording both the SVR and SHI screens simultaneously using a 60 FPS video camera. We then analyzed the recorded video frame by frame to obtain the latency for 12 selections using a stylus (to make it easier to see when the target was selected). We measured two delays:

4- system latency, the time from the stylus entering the target and the system registered selection (i.e., the target circle becomes empty) on the SHI device

- SUI '21, November 09, 10, 2021, Virtual Event, USA
- 4- and transmission latency, the time from when the selection registered on the SHI device was registered on the SVR device (i.e., network latency).

The sum gives the end-to-end latency. We measured latency in this fashion using two sets of devices, a Sony Xperia X as the SHI device with a Samsung Galaxy S10 and Xiaomi Mi 10T Pro as SVR devices for 12 measurements each. The average system latency was 150 ms (SD = 21 ms), and the average transmission latency was 343 ms (SD = 24 ms, min. 317 ms, max. 383 ms). Overall, this delay was noticeable but would likely be lower on a superior VR device (e.g., a dedicated device not using a smartphone).

# 4 METHODOLOGY

We conducted two Fitts' law experiments evaluating our approach. For space reasons, and because the two experiments were similar, we first describe the common methodological elements. We conducted both studies remotely with no physical participant contact due to the COVID-19 pandemic.

The first experiment was intended to determine which of direct or indirect input modes offered better user performance and preference. Past studies have mostly revealed only small and nonsignificant differences in selection performance between direct and indirect input conditions [12, 39]. Thus, a motivating hypothesis was that direct and indirect input would perform comparably; to this end, our analysis uses statistical equivalence testing. Such a result would be promising for future implementations since the indirect mode requires less computing power and battery consumption.

The second experiment also included the WVS technique described above, to determine if input scaling significantly impacts user performance. Previous work indicates that scale factor did not affect selection throughput [9]. However, a smartphone's screen is much smaller than the tablet used in previous work, and onehanded finger input on a smartphone is different from the stylus input on a tablet. Thus, we hypothesized that while low scale factors likely would improve thumb reach without harming performance, higher scale factors (e.g., 2.0 and up) would likely yield significantly lower throughput than a 1-to-1 mapping.

# 4.1 Participants

We recruited 12 participants for both studies. The first study included 6 men and 6 women (aged 19 to 49,  $\mu$  = 30.33,  $\sigma$  = 10.32). One participant was left-handed. In terms of VR experience, 58.3% reported having no VR experience, 41.7% reported several times a year. The second study included 7 men and 5 women (aged 18 to 50,  $\mu$  = 26,  $\sigma$  = 8.16). Two were left-handed, 41.7% had never used VR, 50% used VR several times a year, and 8.3% experienced VR several times a month. Eight participants participated in both studies. Participants were recruited from university students and acquaintances via email. Participants did not receive compensation but were provided with a VR Shinecon (valued at \$15 CAD) that they kept after the study.

<sup>13</sup> https://github.com/Jeffiano/ScreenShareRTC



Figure 4: Study apparatus. Typical setup for the study: a) chair with armrests; b) Smartphone-based VR headset (provided to participants); c) Two Android OS smartphones. d) SVR view of study environment. e) SHI device real view.

## 4.2 Apparatus

Each participant required two smartphones to participate in the studies, one SVR device (i.e., the smartphone acting as a VR display), and one SHI device (i.e., the smartphone for hand input). We provided each participant with a smartphone-based VR head-set, a VR Shinecon Box3D model DV626-9686f (see Figure 4b) to house their SVR device. These devices were ordered online and shipped directly to participants to avoid direct contact during the COVID-19 pandemic. Since the studies were conducted completely remote each participant used a different pair of smartphones as apparatus. We recorded the smartphones used for both the SVR and SHI devices for each participant. All participants used their own standard home-use smartphones ranging from Redmi Note 7 to Samsung Galaxy S20 Ultra. Participants were seated in a chair with armrests to reduce physical strain during the study. Figure 4 depicts a typical setup, required to participate in studies.

During studies, the SVR device was placed inside the VR headset, used as the HMD, and used the integrated back-facing camera to track the SHI device, as described in Section 3. The virtual model of a smartphone showed in a VR environment at the tracked SHI position. The SHI screen feed is also translated to VR environment and rendered in the direct or indirect mode in experiment 1, and always in the indirect mode in experiment 2. Meanwhile, SHI was running the Fitts' law task, recording the participant selection performance data, sending the screen feed to SVR, and rendering a fiducial marker at the upper part of the screen. The VR and realworld first-person view of the system are seen in Figure 4 (d, e).

#### 4.3 Procedure

To minimize contact during the COVID-19 pandemic, we conducted the experiments remotely. We sent the VR headsets to participants via mail. Before starting a study, participants completed consent forms and a demographic questionnaire online. The instructions included detailed how to set up and use the software on smartphones and study details. We used video calls with the participants during the studies to provide extra assistance and answer questions. Figure 5 depicts a participant during the study.

Both studies employed the Fitts' Law reciprocal selection task. The task involved selecting circular targets, placed in a circle, as typical in Fitts' law-based experiments. See Figure 5 (right). Participants were instructed to select the targets as quickly and as accurately as possible. Each sequence started by a double-tap on the SHI smartphone screen. The system tracked the participants' finger position only when their fingers touched the screen. Participants thus had to hold their fingers pressed to the screen at all times during a sequence. The target was considered selected when the cursor's center was placed inside the target circle's border. The timer for the sequence started with the selection of the first target. Targets that were not selected within 10 seconds were considered errors. Regardless if a target was hit or timed out, the next target in the sequence activated. Errors trials were not redone but were excluded from the statistical analysis.

Prior to starting each study, participants did 3 unrecorded practice sequences using direct mode in User Study 1 and using the 1-to-1 mapping in User Study 2. Participants could rest between sequences, as the timer stopped at these points. During these breaks, they could remove the HMD as desired.

Upon completing both studies, participants gave comments on their experience and completed a brief questionnaire. They were then debriefed and reminded to keep the Shinecon HMD as compensation. Each experiment took about 60 minutes, with participants selecting targets in VR for ~45 to 50 minutes. We next describe the specific details for each user study in the following sections.

# 4.4 User Study 1 Experiment Design

User Study 1 used a within-subjects design with two independent variables: Input Mode and Index of Difficulty (ID). The Direct and Indirect input modes are described fully in Section 3.1.

Input Mode: Direct, Indirect;

ID: 1.1, 1.3, 1.5, 1.8, 2.1, 2.4, 2.7, 3.0, 3.3, 3.6



Figure 5: (Left) A participant during the study with a smartphone aligned with the virtual screen in VR. (Right) Close-up of the experiment task presented on the SHI device.

Table 1: Index of Difficulties with their corresponding Amplitudes and Widths.

ID	1.1	1.3	1.5	1.8	2.1	2.4	2.7	3.0	3.3	3.6
Α	115	145	295	400	265	345	220	350	440	445
W	100	100	160	160	80	80	40	50	50	40

The ID values were calculated according to Equation 1 from the following 10 combinations (chosen to fit on the smartphone display) of amplitude and width (in pixels, Table 1):

To counterbalance input mode, half of the participants did direct first, while the other half started with indirect. The study contained 6 blocks, 3 for each condition. Each block had 10 IDs in random order, with one ID per sequence of 15 targets. Each participant completed 6 blocks  $\times$  10 IDs  $\times$  15 targets = 900 selections in total. Over all 12 participants, this yielded 10800 target selections in total or 5400 per condition. The dependent variables included:

**Throughput:** calculated according to the ISO 9241-9 standard and based on Equation 2, in bits per second (bps), where a higher score is better.

**Movement time:** the average selection time, in milliseconds, where a lower score is better.

Although Fitts' law studies commonly also measure error rate, we did not measure this as it was not possible to end a trial with a miss in our experiment.

# 4.5 User Study 1 Results

4.5.1 Throughput. Mauchly's test did not reveal a violation of sphericity in TP data. RM-ANOVA revealed no significant difference for input mode on throughput ( $F_{1,11} = 0.29$ , *ns*). We also conducted a power analysis ( $\mu_1 = 2.109$ ,  $\mu_2 = 2.139$ ,  $\sigma = .45$ ,  $\alpha = .05$ ), and found the power was 0.93 for a 2-sided test. The mean TP difference was just 0.0297 bps. See Figure 6

To test our hypothesis – that there was no difference in direct vs. indirect input – we employed an equivalence test ( $\alpha = .05$ ,  $\delta =$ 



Figure 6: Mean TP by input mode. Error bars show 95% CI.

1 *bps*) since standard statistical ANOVA tests do not determine equivalence, only differences. In other words, a non-significant ANOVA result does not indicate two conditions are equivalent. Thus, for non-significant ANOVA results, we also conducted equivalence testing for Throughput. While this kind of statistical test is not commonly used in HCI studies, it is common in medical studies (e.g., drug comparisons). We used the same indifference zone (1 bps) as previous studies used for a similar non-inferiority test [9, 22]. The choice of 1 bps for an indifference zone was first chosen by Kohli, and the rationale explained in his dissertation [21]: "This value was chosen because a 2004 survey of ISO 9241-9 studies found that the range of throughputs for computer-mouse pointing in five studies was 3.7-4.9bps, a range of 1.2bps [37]. Moreover, in Fitts'



Table 2: Mean TP differences and equivalence test results.

Figure 7: Mean MT by input mode. Error bars show 95% CI.

1954 paper, he labeled 10-12bps, a range of 2bps, as consistent [34]. I conservatively chose a tighter indifference zone of 1bps". We opted to use the same indifference zone based on this argument. Although the indifference zone is based on computer mouse input studies, we use this to define our indifference zone as there are comparatively few Fitts' law studies comparing touch input. An indifference zone of 1 bps indicates that the mean difference between the compared conditions and the bounds of the two-tailed confidence intervals should be lower than 1 bps to be considered equivalent. We used the 2-tailed 90% confidence interval because its equivalent to two one-tailed 95% intervals in opposite directions. Table 2 shows the results of the equivalence test.

Based on this analysis, throughput for the direct and indirect input modes was considered equivalent. This result suggests that throughput is not affected by virtual screen position, in line with our hypothesis, and is consistent with past results on decoupling control/display spaces [11, 12, 22, 39].

4.5.2 Movement time. No violation of sphericity was revealed by Mauchly's test in MT values. RM-ANOVA failed to find a significant difference in movement time between input modes  $(F_{1,11} = 3.18, p = 0.1, ns)$ . See Figure 7

Despite this non-significant difference, we did not conduct the equivalence test for movement time, because, as suggested by Kohli et al. [22], it is not clear what indifference zone for movement time would be reasonable.

4.5.3 Linear Regression Analysis. We conducted a linear regression to model and confirm the linear relation between *MT* and *ID* modeled by Fitts' law for each input mode. As can be seen in Figure 8, both input modes had a strong correlation as expected (i.e.,  $R^2 \approx 0.95$ ). *ID<sub>e</sub>* ranged from 1.37 to 4.69 and 98.5% of the values are in a range from 1.5 to 4.3.

4.5.4 Questionnaire. We asked participants 4 5-point Likert scale questions about their subjective experience. The questions and

Figure 8: Linear regression of Movement time on Index of Difficulty for both conditions. Points represent average MT for all IDs across all participants.

their results (including Friedman analysis comparing the direct to indirect for each question). See Figure 9

#### 4.6 User Study 1 Discussion

The most important result of the study is that the difference in selection throughput between direct and indirect input was both very small and considered statistically equivalent per the result of our equivalence test. This suggests that when using a smartphone in VR, the direct (i.e., real world-like co-located smartphone) condition is not necessary. This is important for VR solutions where tracking methods have high power and processing time requirements; if indirect input works as well as direct input, then such tracking may be unnecessary. For example, when using Cardboard VR, camerabased tracking requires that the camera was always activated, which greatly increases battery drain and smartphone temperature. During our study, 4 participants had their smartphones overheat and needed a 5-minute break to wait for the smartphone to cool down to finish the study. With indirect input, it is possible to use tracking only when the user needs to pick up the smartphone, and tracking can be disabled when the user is otherwise touching the screen.

While most participants were excited about the tracking method, half said that it was sometimes harder to see the virtual screen contents due to tracking problems (e.g., small virtual screen drift). Some noted that indirect was more ergonomic; for long usage, they did not need to lower their head or raise their hand to see the screen. Only one commented that indirect was less natural than direct. Thus, if VR smartphone input is used for a realistic game (e.g., as a game object), direct input may be preferable. Three participants (who were familiar with VR) said it would be a nice feature to be able to use a smartphone during a VR session.





Table 3: Index of Difficulties with their corresponding Amplitudes and Widths.

ID	1.1	1.5	1.8	2.2	2.5	2.9	3.2	3.6
А	115	295	400	290	375	325	410	445
W	100	160	160	80	80	50	50	40

## 4.7 User Study 2 Experiment Design

User Study 2 used a within-subjects design with two independent variables: Scale Factor, and Index of Difficulty (ID).

Scale Factor (SF): 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4;

ID: 1.1, 1.5, 1.8, 2.4, 2.5, 2.9, 3.2, 3.6

*ID* values were calculated according to Equation 1 from the following 8 amplitude/width pairs (in pixels, Table 3):

The SFs were applied to the cursor position using the WVS technique as described earlier. The study contained 8 blocks in total, one SF per block. The order of SFs was counterbalanced according to a balanced Latin square. Within each block, ID order was randomized, with one ID per sequence of 15 targets. In total, there were 8 blocks  $\times$  8 IDs  $\times$  15 targets = 960 selections per participant. Across all 12 participants, this yields 11520 target selections in total.

Our dependent variables included:

**Throughput:** calculated according to ISO 9241-9 and based on Equation 2, in bits per second (bps).

**Movement time:** the average selection time, in milliseconds. Like User Study 1, we did not include an error rate measurement.

#### 4.8 User Study 2 Results

4.8.1 Throughput. Mauchly's test did not reveal a violation of sphericity, thus we employed ANOVA. RM-ANOVA on throughput revealed a significant difference for scale factor ( $F_{7,77} = 8.05$ , p < 0.001). Mean TP values and posthoc pairwise differences test results are seen in Figure 10, where lines and brackets (i.e., "[") indicate pairwise significant differences between scale factors and groups of other scale factors based on Bonferroni corrected posthoc tests. Larger SF values tended to yield lower throughput; participants were more effective with lower SFs. This confirms our hypothesis regarding throughput.

4.8.2 Movement time. RM-ANOVA revealed significant differences for movement time as well. There was a significant main effect of scale factor on movement time ( $F_{7,77} = 3.81$ , p = 0.001). Results indicate that scale factors greater than 2.0 yielded lower selection



Figure 10: Mean TP for each SF value and their pairwise significant differences. Error bars indicate 95% CI.



Figure 11: Mean MT for each SF value and their pairwise significant differences. Error bars indicate 95% CI.

times, confirming our hypothesis. Post-hoc pairwise differences and mean MT scores are seen in Figure 11

SUI '21, November 09, 10, 2021, Virtual Event, USA



Figure 12: Linear regression of Movement time on Index of Difficulty. Points represent average MT for all SF and ID combinations across all participants.

4.8.3 Linear Regression Analysis. A linear regression analysis confirmed the linear relationship between ID and MT modeled by Fitts' law. Figure 12 depicts the regression model with average MT for each SF and ID combination. As expected, the MT and ID relation strongly fit the linear model ( $R^2 = 0.94$ ).  $ID_e$  ranged from 0.75 to 5.1 and 98.3% of the values were between 1 and 4.

4.8.4 *Questionnaire.* We asked participants if they felt the scale factor affected their performance. While answers were distributed from low to high, 33% felt it had a strong influence, and 25% felt it had a medium effect on their performance.

#### 4.9 User Study 2 Discussion

Unlike the past work with stylus and tablet input in VR [9], we found significant differences due to Scale Factor on throughput and movement time. The small SFs (below 1.5) were significantly better than large SFs (1.6 and above). We believe the reason for this difference is the physical size difference of smartphone and tablet touchscreens. Tablets are generally several times larger than smartphones. In our study smartphones were held with a vertical orientation in one hand and input with the same hand's finger, which made the actual interactive zone even smaller. However, we used the same SFs as those used with a tablet in the previous study. Thus, our results suggest the maximum SF which can be applied to smartphone input without significant influence on performance appears to be roughly 2.0. This means VR smartphone input can be scaled to cover twice the size of the thumb-zone with only a small (albeit significant) drop in efficiency. Even for large "phablet" smartphones (e.g., around 6 inches) the thumb-zone usually covers at least half of the screen. A 2.0 SF means that a user could comfortably reach targets across the entire smartphone screen using one-hand thumb input. That said, it may cast doubt on the applicability of such scaled input methods with very small touchscreen devices, e.g., smartwatches, in VR. This is a topic for future study.

While all the participants noticed some changes in input with high SF, some could not explain the difference without a hint. Two participants noted small SFs helped them reach targets on the edge of the smartphone. They said: "During some conditions, I dragged my finger on smaller distances to hit the targets, so I didn't need to struggle and stretch my thumb to select the ones that were far away, it was helpful".

# 5 REAL APP INTEGRATION PROOF OF CONCEPT

To demonstrate how our tracking method could be used with "real" applications (i.e., general-purpose smartphone use in VR) we developed a proof-of-concept prototype. The key challenge was to be able to display the entire smartphone screen in VR, despite requiring a fiducial marker for tracking in the direct input mode. If using the indirect mode, this is easy to implement (i.e., no fiducial marker is necessary) and the RTC app can stream the screen regardless of the active application. However, the task is more complicated if tracking is required, i.e., to support co-locating the virtual with the real smartphone in direct input mode.

The Android OS supports putting the view on top of all running apps and allows touch input to "go through" the view. We employed both features to display the fiducial marker for tracking. However, at the time of writing, the OS does not support capturing the screen feed without one particular view, i.e., to exclude the fiducial marker and get only the screen feed underneath. To resolve this, we made the fiducial marker view blink as fast as possible (e.g., on alternating frames), then filtered the frames to exclude those with the fiducial marker before sending them to SVR. While this workaround allows real apps to run on a tracked smartphone in VR, it comes at a cost; the virtual screen frame rate is cut in half and the SHI smartphone tracking delay doubles. The prototype is seen in Figure 13, which depicts multiple Android apps running in direct input mode.

# 6 OVERALL DISCUSSION & CONCLUSIONS

We proposed a method for tracking and using a smartphone as an input device in VR, and presented two studies evaluating our solution. At roughly 2 bps, the performance of smartphone-based target selection in VR is about half that of mouse-based desktop input [37], but similar to other 3D selection methods [41, 43]. The well-known performance cost of latency likely influenced the results as well [30, 40]. Our studies also reveal that the actual position of the smartphone screen need not be co-located with the actual tracked smartphone. In User Study 1, per equivalence statistical testing, performance was statistically the same between our direct and indirect conditions. In other words, actual 6DOF tracking of the smartphone may not be necessary at all times, and could be disabled, e.g., between grasps. Our second study determined that there are limits on how much scaling can be applied before users notice or are affected negatively by it. The threshold, in our experiment, seems to be around a scale factor of 1.6. Performance degraded for scale factors higher than this.

Overall, our results give insight into the use of smartphones in VR. We argue that smartphones are interesting platforms as an interaction device in VR, providing tactile feedback (that can be virtually scaled, to an extent) and a high-precision touchscreen. Future work will investigate employing our method in more generalpurpose smartphone tasks, rather than controlled experiment tasks like selection, and determining the maximum scale thresholds on even smaller touchscreens (e.g., smartwatches).



Figure 13: The prototype with multi-app support with device tracking. The fiducial marker is continuously blinking, but completely absent on the virtual screen. (Left) The Android home screen. (Right) Android calculator app running in VR.

#### REFERENCES

- Haley Adams, Gayathri Narasimham, John Rieser, Sarah Creem-Regehr, Jeanine Stefanucci, and Bobby Bodenheimer. 2018. Locomotive recalibration and prism adaptation of children and teens in immersive virtual environments. *IEEE Transactions on Visualization and Computer Graphics* 24, 4: 1408–1417. https://doi.org/10.1109/TVCG.2018.2794072
- [2] Tatsuya Amano, Shugo Kajita, Hirozumi Yamaguchi, Teruo Higashino, and Mineo Takai. 2018. Smartphone applications testbed using virtual reality. ACM International Conference Proceeding Series: 422–431. https://doi.org/10.1145/3286978. 3287028
- [3] Mahdi Azmandian, Mark Hancock, Hrvoje Benko, Eyal Ofek, and Andrew D. Wilson. 2016. Haptic retargeting: Dynamic repurposing of passive haptics for enhanced virtual reality experiences. *Conference on Human Factors in Computing Systems - Proceedings*: 1968–1979. https://doi.org/10.1145/2858036.2858226
- [4] Huidong Bai, Li Zhang, Jing Yang, and Mark Billinghurst. 2021. Bringing Fullfeatured Mobile Phone Interaction into Virtual Reality. *Computers & Graphics* 97: 42–53. https://doi.org/10.1016/j.cag.2021.04.004
- [5] Bobby Bodenheimer, Sarah Creem-Regehr, Jeanine Stefanucci, Elena Shemetova, and William B. Thompson. 2017. Prism aftereffects for throwing with a selfavatar in an immersive virtual environment. *Proceedings - IEEE Virtual Reality* 0: 141–147. https://doi.org/10.1109/VR.2017.7892241
- [6] Jean Botev, Joe Mayer, and Steffen Rothkugel. 2019. Immersive mixed reality object interaction for collaborative context-aware mobile training and exploration. Proceedings of the 11th ACM Workshop on Immersive Mixed and Virtual Environment Systems, MMVE 2019: 4–9. https://doi.org/10.1145/3304113.3326117
- [7] David Brickler, Robert Teather, Andrew Duchowski, and Sabarish Babu. 2020. A Fitts' Law Evaluation of Visuo-haptic Fidelity and Sensory Mismatch on User Performance in a Near-field Disc Transfer Task in Virtual Reality. ACM Transactions on Applied Perception 17: 1–20. https://doi.org/10.1145/3419986
- [8] Sungmin Cho, Won-Seok Kim, Seo Hyun Park, Jihong Park, and Nam-Jong Paik. 2020. Virtual Prism Adaptation Therapy: Protocol for Validation in Healthy Adults. Journal of visualized experiments: JoVE, 156. https://doi.org/10.3791/60639
- Seyed Amir Ahmad Didehkhorshid and Robert J. Teather. 2020. Selection performance using a scaled virtual stylus cursor in VR. Proceedings Graphics Interface 2020-May.
- [10] P M Fitts. 1992. The information capacity of the human motor system in controlling the amplitude of movement. 1954. *Journal of experimental psychology. General* 121, 3: 262–269. https://doi.org/10.1037//0096-3445.121.3.262
- [11] Clifton Forlines. 2007. Direct-Touch vs . Mouse Input for Tabletop Displays. 647–656.
- [12] Clifton Forlines, Mitsubishi Electric, and Ravin Balakrishnan. 2008. Evaluating Tactile Feedback and Direct vs. Indirect Stylus Input in Pointing and Crossing Selection Tasks.
- [13] Akira Fuji, Makoto Fujimura, and Toshio Higashi. 2018. Virtual Environment of Prism Adaptation for Unilateral Spatial Neglect. 2018 IEEE 7th Global Conference on Consumer Electronics, GCCE 2018: 81–82. https://doi.org/10.1109/GCCE.2018. 8574783
- [14] Makoto Fujimura, Kazuya Nakatomi, Akira Fujii, and Toshio Higashi. 2019. Problem of pointing target in virtual environment for prism adaptation. 2019 IEEE 1st Global Conference on Life Sciences and Technologies, LifeTech 2019: 8–9. https://doi.org/10.1109/LifeTech.2019.8884066
- [15] Sascha Gebhardt, Sebastian Pick, Thomas Oster, Bernd Hentschel, and Torsten Kuhlen. 2014. An evaluation of a smart-phone-based menu system for immersive virtual environments. *IEEE Symposium on 3D User Interfaces 2014, 3DUI 2014 -Proceedings*: 31–34. https://doi.org/10.1109/3DUI.2014.6798837
- [16] Dustin T. Han, Mohamed Suhail, and Eric D. Ragan. 2018. Evaluating remapped physical reach for hand interactions with passive haptics in virtual reality. *IEEE Transactions on Visualization and Computer Graphics* 24, 4: 1467–1476. https://doi.org/10.1016/j.com/10016/j.com/10016/j.com/10016/j.com/10016/j.com/10016/j.com/10016/j.com/10016/j.com/10016/j.com/10016/j.com/10016/j.com/10016/j.com/10016/j.com/10016/j.com/10016/j.com

//doi.org/10.1109/TVCG.2018.2794659

- [17] John Paulin Hansen, Vijay Rajanna, I. Scott MacKenzie, and Per Bækgaard. 2018. A Fitts' law study of click and dwell interaction by gaze, head and mouse with a head-mounted display. *Proceedings - COGAIN 2018: Communication by Gaze Interaction*: 2–6. https://doi.org/10.1145/3206343.3206344
- [18] Samory Houzangbe, Geoffrey Gorisse, Olivier Christmann, and Simon Richir. 2018. Integrability and reliability of smart wearables in virtual reality experiences: A subjective review. ACM International Conference Proceeding Series. https://doi. org/10.1145/3234253.3234305
- [19] Samat Imamov, Daniel Monzel, and Wallace S Lages. 2020. Where to display? How Interface Position Affects Comfort and Task Switching Time on Glanceable Interfaces. 851–858. https://doi.org/10.1109/VR46266.2020.00110
- [20] Nicholas Katzakis, Robert J. Teather, Kiyoshi Kiyokawa, and Haruo Takemura. 2015. INSPECT: extending plane-casting for 6-DOF control. *Human-centric Computing and Information Sciences* 5, 1. https://doi.org/10.1186/s13673-015-0037-y
- [21] Luv Kohli. 2013. Redirected Touching. Retrieved from http://www.cs.unc.edu/ techreports/13-002.pdf
- [22] Luv Kohli, Mary C. Whitton, and Frederick P. Brooks. 2012. Redirected touching: The effect of warping space on task performance. *IEEE Symposium on 3D User Interfaces 2012, 3DUI 2012 - Proceedings*: 105–112. https://doi.org/10.1109/3DUI. 2012.6184193
- [23] Luv Kohli, Mary C. Whitton, and Frederick P. Brooks. 2013. Redirected Touching: Training and adaptation in warped virtual spaces. *IEEE Symposium on 3D User Interface 2013, 3DUI 2013 - Proceedings*: 79–86. https://doi.org/10.1109/3DUI.2013. 6550201
- [24] Alena Kovarova and Maros Urbancok. 2014. Can virtual reality be better controlled by a smart phone than by a mouse and a keyboard? ACM International Conference Proceeding Series 883: 317–324. https://doi.org/10.1145/2659532.2659608
- [25] Kapil Kumar, Arindam Mondal, and Gaurav Gupta. 2019. Watch360: A Device to Enable and Detect Tilt, Translation and Rotation of a Watch Bezel. 2019 16th IEEE Annual Consumer Communications and Networking Conference, CCNC 2019. https://doi.org/10.1109/CCNC.2019.8651875
- [26] Hai Ning Liang, Yuwei Shi, Feiyu Lu, Jizhou Yang, and Konstantinos Papangelis. 2016. VRM controller: An input device for navigation activities in virtual reality environments. Proceedings - VRCAI 2016: 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry 1: 455–460. https: //doi.org/10.1145/3013971.3014005
- [27] Kent Lyons. 2016. 2D input for virtual reality enclosures with magnetic field sensing. International Symposium on Wearable Computers, Digest of Papers 12-16-Sept: 176–183. https://doi.org/10.1145/2971763.2971787
- [28] I. Scott MacKenzie and Poika Isokoski. 2008. Fitts' throughput and the speedaccuracy tradeoff. Conference on Human Factors in Computing Systems - Proceedings: 1633–1636. https://doi.org/10.1145/1357054.1357308
- [29] I Scott MacKenzie. 1992. Fitts' Law as a Research and Design Tool in Human-Computer Interaction. *Hum.-Comput. Interact.* 7, 1: 91–139. https://doi.org/10. 1207/s15327051hci0701\_3
- [30] I Scott MacKenzie and Colin Ware. 1993. Lag as a Determinant of Human Performance in Interactive Systems. In Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems (CHI '93), 488–493. https://doi.org/10.1145/169059.169431
- [31] Anne Collins Mclaughlin, Wendy A Rogers, and Arthur D Fisk. 2009. Using Direct and Indirect Input Devices: Attention Demands and Age-Related Differences. 16, 1: 1–15. https://doi.org/10.1145/1502800.1502802
- [32] Tim Menzner, Travis Gesslein, Alexander Otte, and Jens Grubert. 2020. Above Surface Interaction for Multiscale Navigation in Mobile Virtual Reality. Proceedings - 2020 IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2020: 372–381. https://doi.org/10.1109/VR46266.2020.1581107639032
- [33] Peter Mohr, Markus Tatzgern, Tobias Langlotz, Andreas Lang, Dieter Schmalstieg, and Denis Kalkofen. 2019. TrackCap: Enabling smartphones for 3D interaction

on mobile head-mounted displays. Conference on Human Factors in Computing Systems - Proceedings: 1–11. https://doi.org/10.1145/3290605.3300815

- [34] Paul M. Fitts. 1954. the Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *Journal of Experimental Psychology* 47, 6: 381–391. Retrieved from http://www2.psychology.uiowa.edu/faculty/mordkoff/ InfoProc/pdfs/Fitts 1954.pdf
- [35] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. 1996. The go-go interaction technique. 79–80. https://doi.org/10.1145/237091.237102
- [36] Boris Smus and Christopher Riederer. 2015. Magnetic input for mobile virtual reality. ISWC 2015 - Proceedings of the 2015 ACM International Symposium on Wearable Computers: 43–44. https://doi.org/10.1145/2802083.2808395
- [37] R. William Soukoreff and I. Scott MacKenzie. 2004. Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. International Journal of Human Computer Studies 61, 6: 751–789. https: //doi.org/10.1016/j.ijhcs.2004.09.001
- [38] R J Teather, D Natapov, and M Jenkin. 2010. Evaluating haptic feedback in virtual environments using ISO 9241–9. In 2010 IEEE Virtual Reality Conference (VR), 307–308. https://doi.org/10.1109/VR.2010.5444753
- [39] Robert J Teather, Robert S Allison, and Wolfgang Stuerzlinger. 2009. Evaluating Visual / Motor Co-location in Fish-Tank Virtual Reality.
- [40] Robert J Teather, Andriy Pavlovych, Wolfgang Stuerzlinger, and I Scott MacKenzie. 2009. Effects of Tracking Technology, Latency, and Spatial Jitter on Object

Movement. In *Proceedings of the 2009 IEEE Symposium on 3D User Interfaces* (3DUI '09), 43–50. https://doi.org/10.1109/3DUI.2009.4811204

- [41] Robert J Teather and Wolfgang Stuerzlinger. 2013. Pointing at 3d Target Projections with One-Eyed and Stereo Cursors. In *Proceedings of the SIGCHI Conference* on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, 159–168. Retrieved from https://doi.org/10.1145/2470654. 2470677
- [42] Robert Teather and I MacKenzie. 2020. Position vs. Velocity Control for Tilt-Based Interaction. . 51–58. https://doi.org/10.1201/9781003059325-7
- [43] Robert Teather and Wolfgang Stuerzlinger. 2011. Pointing at 3D targets in a stereo head-tracked virtual environment. https://doi.org/10.1109/3DUI.2011.5759222
- [44] Philipp Tiefenbacher, Steven Wichert, Daniel Merget, and Gerhard Rigoll. 2014. Impact of coordinate systems on 3D manipulations in mobile augmented reality. ICMI 2014 - Proceedings of the 2014 International Conference on Multimodal Interaction: 435–438. https://doi.org/10.1145/2663204.2663234
- [45] André Zenner, Hannah Maria Kriegler, and Antonio Krüger. 2021. HaRT The Virtual Reality Hand Redirection Toolkit. Conference on Human Factors in Computing Systems - Proceedings. https://doi.org/10.1145/3411763.3451814
- [46] Li Zhang, Huidong Bai, Mark Billinghurst, and Weiping He. 2020. Is This My Phone? Operating a Physical Smartphone in Virtual Reality. SIGGRAPH Asia 2020 XR, SA 2020, Figure 1: 1–2. https://doi.org/10.1145/3415256.3421499