Viewport- and World-based Personal Device Point-Select Interactions in the Augmented Reality

by

Yuan Chen

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2020

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see **Statement of Contributions** included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis includes first-authored content from the following conference publication:

- Yuan Chen, Keiko Katsuragawa, and Edward Lank. 2020. Understanding Viewport- and World-based Pointing with Everyday Smart Devices in Immersive Augmented Reality. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–13. DOI:https://doi.org/10.1145/3313831.3376592

The content from this paper has been adapted and extended for this thesis.

Abstract

Personal smart devices have demonstrated a variety of efficient techniques for pointing and selecting on physical displays. However, when migrating these input techniques to augmented reality, it is both unclear what the relative performance of different techniques will be given the immersive nature of the environment, and it is unclear how viewport-based versus world-based pointing methods will impact performance. To better understand the impact of device and viewing perspectives on pointing in augmented reality, in this thesis, we present the results of two controlled experiments comparing pointing conditions that leverage various smartphone- and smartwatch-based external display pointing techniques and examine viewport-based versus world-based target acquisition paradigms. Our results demonstrate that viewport-based techniques offer faster selection and that both smartwatch- and smartphone-based pointing techniques represent high-performance options for performing distant target acquisition tasks in augmented reality.

Acknowledgements

I would like to thank all the little people who made this thesis possible.

First, I would like to thank my amazing supervisors, Edward Lank and Keiko Katsuragawa for their support, guidance and endless patience throughout the past two years. I would not be able to achieve my thesis without them.

Second, I would like to thank Daniel Vogel and James Wallace for being my thesis readers and providing valuable feedback for both my thesis and my future research.

I feel grateful to meet many talented people while working in the HCI lab at Waterloo. Thank you to everyone in the lab who helped and supported me: Sangho, Louis, Johann, Nikhita, Matthew, Hemant and Jeremy; to Ed's group: Damien, Rina, Jay, Caesar, Bahar and Anastasia; and to everyone else in/outside the lab: Yangtian, Peng, and Kaisong for being wonderful friends.

To all friends: thank you for your support throughout these years. Thanks to Lyc and Lyh for being my friend since middle school and making me feel connected to home. Thank you to Gqi, Cbai, Xduac, Wls, Hcai and Dbs for being my buddies. Special thanks to Meixin for her accompany and support.

Finally, I would like to thank my parents for their love and support. I would not be where I am today without them.

Dedication

To Mom and Dad.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Pointing and selecting are foundational aspects of interaction both on physical displays [20, 52, 53, 61] and in virtual (VR) and augmented reality (AR) [3, 10, 17, 24, 37, 79]. In AR, the importance of pointing is motivated by the need to interact with an increasingly augmented world: to interact with increasingly pervasive augmentation, we need to select from among available augmentations prior to interaction with the desired augmentation. Pointing is an effective mechanism to perform this selection.

In recent work, Siddhpuria et al. [61] contrasted a set of different personal device point-select techniques to support interaction with external displays positioned at greater-than-arms' length. The rationale for using personal devices was that, rather than requiring users to bring specialized controllers or restricting interaction to spaces equipped with motion tracking, a personal device could serve as a proxy to support interaction in a wide variety of contexts [31, 54, 55].

Our initial goal was to replicate aspects of the Sidhipuria et al.'s [61] study in AR environments because, in these environments, it is also often the case that users must interact with distant projected content. While personal devices have frequently been used to perform manipulations, including pointing, in AR environments [4, 6, 32], we are aware of no work that explicitly compares different personal device point-select interactions.

However, a challenge presented itself when adapting personal device point-select interactions to AR. While distant pointing and selecting are common tasks within AR (and VR) environments, techniques for pointing and selecting remain fragmented across head-mounted displays (HMDs). The use of specialized controllers, specialized gestures, and varied positional tracking technologies all contribute to this fragmentation. For example, to efficiently interact with virtual objects in HMD environments, users either use specialized controllers (e.g. HTC Vive [29], Oculus Rift S [49]) or perform gestures (e.g. Microsoft Hololens [44]). In the first scenario, specialized controllers are independent of the users' head movement, but require additional hardware resources for position tracking. In the second scenario, the sensor is attached to the HMD and captures gestures directly in front of the HMD. Finally, it is also common to see gaze-based pointing techniques, where the cursor is centred on the user's field of view, to support pointing without specialized controllers or front-of-HMD sensor input.

More broadly, the above input techniques can be segmented into two broad input metaphors for HMD environments: the *Exocentric* or *world-based* metaphor and the *Egocentric* or *viewport-based* metaphor [7]. Simply, the world-based metaphor allows the cursor to move independently of the movement of the HMDs or, more specifically, the HMDs' field-of-view. In contrast, the viewport-based metaphor places the cursor within the viewport of the HMDs and the cursor moves with HMD movement. Stated another way, you must either move the cursor around the environment to a new location, i.e. the cursor must traverse the external world, or, you must move the viewport to the location of the object of interest and then point within the viewport. We adopt viewport- and world-based metaphors throughout this thesis to classify these two pointing metaphors.

Recent studies in AR have leveraged the viewport-based metaphor [10, 24, 37], where users complete all tasks with the cursor within the view. In contrast, in VR, the world-based metaphor is commonly used for target selection [3, 17, 79]. Though emerging research exploring and developing interactive techniques is continually occurring in HMD-based AR and VR, to the best of our knowledge in examining the research literature, it remains unclear how users' viewing metaphors (viewport vs world) affect their input performance in either augmented or virtual reality. While the focus of our work is AR environments, the common HMD paradigm across AR and VR argues that either one of these two paradigms, viewport or world, might prove more effective for pointing.

Given the above viewing perspective confound, this comparative work independently explores the use of two different personal devices, a smartphone and a smartwatch, to support pointing and selecting in AR environments. For each device, we explore different point-select techniques across two different pointing metaphors: world-based and viewport-based. We find that the viewport-based paradigm provides faster selection than the world-based paradigm without sacrificing targeting accuracy. We also demonstrate that modern smart devices are an effective input device for AR environments. Finally, based on our results, we provide suggestions for devices and techniques in HMD-based AR environments.

1.1 Contributions

This work contributes to three aspects:

1. contributes empirical findings that point-select techniques using smart devices that leverage viewport-based viewing are faster, as accurate and require less workload.
2. provides an easy-to-implement and detailed description of pointing techniques using smart devices in augmented reality.
3. proposes design guidelines for developing pointing techniques using smart devices based on different augmented reality environments.

1.2 Outline

This thesis is organized as follows:

- Chapter 2 summarizes prior work and provides a basic background to the pointing techniques in both physical and virtual world.
- Chapter 3 introduces the description of pointing paradigms (viewport-based and world-based), pointing techniques using smartphones and smartwatches and related selection methods.
- Chapter 4 describes the general study setup, experimental design and system implementation in this work.
- Chapter 5 describes the first experiment and its results, in which we tested different pointing conditions using smartphones.
- Chapter 6 describes the second experiment and its results, in which we examined different pointing conditions using smartwatches.
- Chapter 7 discusses the design implications as well as the future work surrounding pointing in VR/AR and techniques using smart devices.
- Chapter 8 concludes by summarizing our work.

Chapter 2

Literature and Related Work

2.1 Using Fitts' Law to Model Target Acquisition

Fitts' Law [18] is the most commonly used approach to study new target acquisition techniques to complete an spatial acquisition task. It links movement time to the concept of the index of difficulty (ID):

$$MT = a + bID, \text{where } ID{=}log_2(\frac{2D}{W})$$

a and b are empirically determined regression coefficients and the logarithmic term ID is a function of both the distance D and target width W. Motivated by Shannon's theorem, MacKenzie [39] proposed another formulation of ID:

$$ID = log_2(\frac{D}{W} + 1)$$

Both formulations suggest that Fitts' Law accounts for the movement time by target distance and width: larger target distance and smaller target width caused longer selection time.

Fitts' Law was initially proposed to model stationary 1D target selection in physical world, which has been extended to accommodate various pointing scenarios. Jagacinski et al. [30] introduced target moving speed into the Fitts' Law to model the movement time in a moving target acquisition task:

$$MT = d + cD + b(V + 1)(\frac{1}{W} - 1)$$

MacKenzie and Buxton [40] proposed the SMALLER-OF model:

$$MT = a + blog_2(\frac{D}{min(W, H)} + 1)$$

and W' model:

$$MT = a + blog_2(\frac{D}{W'} + 1)$$

W' is in the direction of motion and extended Fitts' Law into 2D pointing tasks. However, several problems existed with these two models. $ID_{W'}$ ignored the directional constraint of a target while movement angle and interchanging target width and height have no effect on ID_{min}. Accot and Zhai [1] examined the the effect of a target's width and height ratio with a bivariate pointing model to address these problems:

$$MT = a + blog_2(\sqrt{(\frac{D}{W})^2 + \eta(\frac{D}{H})^2} + 1)$$

These extended models have contributed to many applications of Fitts' Law in graphical user interfaces, such as cursor augmentation [22], widgets expansion [41] and point-drag items [60].

While looking at 3D environments, Ware and Balakrishnan [74] noted that the SMALLER-OF model could be easily extended into 3D:

$$MT = a + blog_2(\frac{D}{min(W, H, Z)} + 1)$$

Grossman and Balakrishnan [21] evaluated multiple Fitts' Law models and studied effects of the target size and movement angle on pointing performance in a 3D volumetric space. They noted that target size along the primary movement axis of the input technique has greater impacts than other two dimensions. Also, inspired by the bivariate model[1], their weighted trivariate pointing models better fitted data than variants of SMALLER-OF models:

$$MT = a + blog_2(\sqrt{(\frac{D}{W})^2 + \alpha(\frac{D}{H})^2 + \beta(\frac{D}{Z})^2} + 1)$$

Murata and Iwase [46] found that an extended model on Fitts' Law with a directional parameter better accounted for pointing performance in the real-world three-dimensional pointing while Wingrave and Bowman [77] found that the conventional Fitts' law still holds on 3D pointing tasks using raycasting in virtual environments. Similarly, Teather and Stuerzlinger [69] conducted a series of experiments to examine both 2D and 3D pointing tasks in fish-tank virtual environment and found that the movement time performance of these tasks could be sufficiently well-modelled by conventional Fitts' Law.

In this thesis, we use conventional Fitts' Law in MacKenzie's formulation [39] as an effective tool to examine our pointing techniques using smart devices and reveal how the target distance and width can impact the performance of these techniques in the augmented reality environment.

2.2 Viewport- and World-based Pointing

Numerous techniques based on either viewport-based or world-based pointing paradigms have been proposed for 3D target selection in both physical (e.g. large display) and virtual (e.g. AR/VR) environments [3]. The most basic form of viewport-based target acquisition is gaze-based pointing [2, 37], where the user acquires a target by positioning a fixed point within their field-of-view over the target and then performs some action to select. Most of prior work have focused on integrating and coordinating gaze-based input pointing with other input modalities, such as the mouse [70, 80], keyboards [36, 66], handheld controllers [37] and the touch input [64, 66, 65]. A prominent motivation for the integration and coordination is to place a cursor to the vicinity of a user's point-of-regard such that manual fine adjustments can be performed with other inputs, which contributes to speeding up the target selection time and reducing users efforts with either external displays [80] or in cluttered virtual environments [2, 37].

Alongside this basic form of gaze-based input, target acquisition is performed directly using either a specialized controller (e.g. Oculus) or the bare hand (e.g. Hololens). When the target is proximal to users, users can directly touch targets around the body with the controllers or the hand [57]. For more distant targets, users can use various forms of raycasting to point, where a ray emits from either a moving-origin (controller or hand) or a fixed-origin (camera, eye gaze location) to select remote targets [3, 16, 63, 78, 2, 5] or grab targets beyond arms' reach with a non-linear mapping function to magnifies the movement of the virtual hands [58, 75]. In addition, several work investigated how different facilitation techniques, such as depth control and occlusion management [3], can improve the efficiency of controllers and raycasting techniques in more complicated virtual environments. For instance, several techniques allow manual adjustment of depth of the raycasting cursor [59]; for example, Bowman et al. [11] leverage a "fishing reel" metaphor and the depth ray technique [23] and its variants [72] leverage 3D movement of a tracking device to control depth: the user moves their hand closer or farther from their body to dynamically adjust the depth of the raycasting cursor. The flexible pointer [50] and sticky-ray [62] allow users to point at partially occluded targets and stick to a hit object with a curved ray. Finally, the iSith metaphor [78] allows bimanual object selection and manipulation. In terms of dense environments, iterative refinement techniques, such as SQUAD[35] and Expand [14] rearranges and filters content to support rapid raycasting selection. Poupyrev et al. [56] classified these selection techniques into egocentric and exocentric metaphors. With exocentric pointing, users interact with the environment from the outside, such as World-In-Miniature technique [67] and volumetric interaction [21], while with egocentric pointing, users interact with the environment from the inside, such as techniques above. This classification is slightly different from our input metaphors. As noted above, these target acquisition approaches can be generally classified into two categories based on whether the cursor/interactor stays within the field-of-view and moves with head movement to remain in the field-of-view (viewport-based) [10, 24, 37] or if the cursor moves independently of the head (world-based) [3, 79, 17].

Alongside straightforward implementations of these two input paradigms, Kytö et al. [37] discussed and evaluated eye gaze, head pointing and several multi-model techniques using gaze and head, which pre-assumed that users are aware of the cursor and can move it with head rotation. Researchers have also explored various forms of around-body, eyes-free selection [79]. However, despite the application of viewport- and world-based input paradigms across AR/VR [15, 17], we are aware of no research that explicitly compares these two viewing perspectives.

2.3 Distant Pointing Techniques with Smart Devices in AR/VR

Smart devices, particularly smartphones and smartwatches, have provided convenient interaction methods to point at remote targets on external displays [8, 25, 31, 55] with either their interactive touch surfaces or computationally powerful built-in sensors, such as the gyroscope and accelerometer sensors. Abundant point-select techniques on smart devices enable rich interaction space with both distant large displays and virtual environments. For instance, Stellmach and Dachselt [65] explore the interaction space between smartphones and distant displays to support seamless object selection, position and manipulation. Büschel et al. [13] investigate smartphones as interaction controller to support zoom and pan in a AR HMD environment. The most recent work from Zhu and Grossman [81] presents a design space of cross-device interactions between smartphones and AR head-mounted displays: users can interact with the same virtual object in both a 2D display and a 3D HMD augmented reality environment.

It is tempting to assume that results from recent comparative studies of external display smart device pointing techniques [61] can be directly applied to augmented reality environments, but we hesitate to assume that this is the case. In Siddhpuria et al.'s work [61], smartphone-based absolute and relative techniques had similar performance, and the most effective techniques were those which used the display screen as a touchpad to move the cursor. In AR/VR, because targets can be located to the user's side or behind the user [79], touchpad style input might require a significant number of clutch operations, slowing input. As well, the restricted field-of-view of HMD-based AR might simplify absolute mapping. Furthermore, the 3D perspective required during interaction could influence accelerometer-based tracking versus touch-screen tracking in unanticipated ways.

Despite this uncertainty, leveraging personal devices for AR-based manipulations continues to be an attractive area of research. Recent work [4, 6, 45] has developed various techniques using a smartphone, e.g. tracking its pose or using other built-in or added sensors, to provide 6DOF interaction across AR and VR contexts. However, while myriad research explores targeting and personal device interactions in AR/VR, we are aware of no previous work that has explicitly compared different types of pointing techniques using various personal devices and discussed their applicability to AR.

Chapter 3

Methodologies

Our study aims to explore the use of smart devices – smartphones and smartwatches – as a platform of convenience for pointing and selecting tasks in augmented reality. To define a series of techniques, we vary two factors: the pointing paradigm (viewport-based vs world-based) and smart device input techniques that map either touchscreen or IMU onto cursor movement to support pointing.

3.1 Pointing Paradigm: Viewport-based vs World-based

Users see virtual objects in the AR environment through a rectangular *viewport* powered by their headset. In this section, we define three viewing conditions: *Viewport Center* (VC), *Viewport Boundary* (VB) and *World* (WD).

3.1.1 Viewport Center: Viewport-based Control Condition

Viewport Center (VC) is a commonly used solution for target pointing and selection in many off-the-shelf HMDs. In VC, the virtual cursor is always located at the center of the viewport and its movement corresponds to a user's head rotation. VC is also a *Head-only + device* technique [37] where cursor movement relies on head movement and external devices are used only for selection.

VC is considered a baseline interaction for comparison in this study. It is independent of input device used, functions with all HMDs, and exhibits good performance [37].

3.1.2 Viewport Boundary: Viewport-based Experimental Condition

Recent studies have shown that, compared with traditional hand-based methods, eye-only selection limits the user's ability to recall the environment [68] and causes high error rates [37]. *Viewport Boundary* (VB) is a refined viewing condition that seeks to mitigate these issues. VB allows a user to scan the environment, supporting environmental recall, while also allowing cursor movement within the viewport for target selection.

We performed a series of pilots of pointing techniques within the AR space. During the interaction, users expected the cursor to respond to head movement when they moved the head a sufficiently large distance, but for small head movements, they assumed that the cursor would stay under input device control. Alongside this, during these pilots, we noted that if a cursor moved close to the edges of the viewport, it would become dimmed and participants would lose track of the cursor, a result of the offset location making content harder to find. However, when the cursor was in the middle of the display, participants had less trouble tracking the cursor. These observations – the tendency to lose the cursor at the edges and the need for cursor stability versus head movement when acting within the center of the field of view – drove the design of our VB condition.

In the context of our VB condition, the *Boundary* is a band around the edge of the display. We tuned this boundary around the display during pilot studies; because binocular human vision captures in a wider horizontal than vertical range [24], the upper and lower sides of the *boundary* were 10% of viewport's height from top and bottom edges of the viewport respectively and 20% of the viewport's width from the left and right edges of the viewport.

Interaction in VB proceeds as follows. When a user's head is stable, a cursor inside the display can be repositioned by manipulating an input device to control the cursor. When the head moves, if the cursor is within the *Boundary* but outside the trailing boundary, i.e on the side of *Boundary* that is moving toward the cursor during head movement, the cursor remains fixed in position. However, if the head moves sufficiently far that the trailing *Boundary* contacts the cursor, the cursor is 'pushed' in the direction of head movement, such that it stays within the *Boundary*. This design both reduces users' effort in tracking the cursor when moving the head a sufficiently large distance and enables full use of the viewport while pointing.

3.1.3 World: World-based Experimental Condition

World (WD) is a world-based paradigm commonly used in VR. In the WD condition, cursor movement is independent of head movement and only relies on input devices to control displacement.

The cursor is initially generated at the center of the virtual wall in AR. To relocate the cursor to a target location outside the field of view, the input device must be used to reposition the cursor (it does not follow the user's gaze). This interaction paradigm is most similar to the point-select paradigm of large public display or computer display pointing interactions.

3.1.4 Pointing Paradigm Summary

Figure 3.1 summarizes three viewing conditions: a red sphere (target) and four white spheres (cursors) are on the virtual wall, with three cursors within the viewport: (a) *Viewport Center*: cursor is always located at the center of the viewport and its movement only depends on the head movement. (b) & (c) *Viewport Boundary*: a cursor will move along with the head movement when it hits the boundary (dashed line) but it can move outside of the boundary when it is manipulated by smartphones and smartwatches. (d) *World*: cursor movement relies only on input devices. VC and VB are both an viewport-based pointing paradigm while WD is an world-based pointing paradigm.

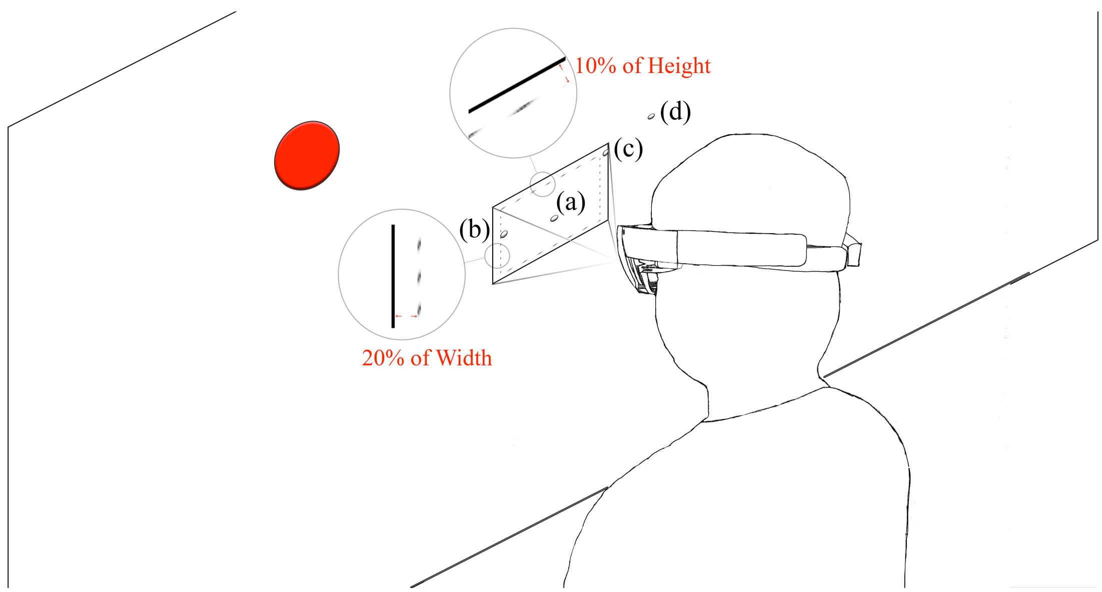

Figure 3.1: Illustration of three viewing conditions.

3.2 Pointing Techniques

Within the space of personal device pointing techniques, Siddhpuria et al. discussed three different types of pointing techniques [61]: *Relative Touch-based* (TR), where the screen of the device serves as a touchpad for repositioning an external cursor and capturing clicks; *Absolute Touch-based* (TA), where, again, the touchscreen captures input, but using an absolute mapping of touchscreen to external display; and *Relative Rotation-based* (RR) techniques, which use the on-board IMU to capture pitch and yaw movement of the smart device.

We implemented five pointing techniques from [61], three on smartphone and two on smartwatch, as shown in Table 3.1. We include two-handed relative touch and one handed relative rotation techniques on each device. We also include two-handed absolute touch input on the smartphone, but not the smartwatch, because, while feasible [28], it is not practical to absolutely map between such different sizes of display. Table 3.1 leverages terms from [61] to describe the pointing techniques for consistency; using this terminology, we label techniques with one letter Device-Handedness-Input-Mapping monikers: P2TA (Phone, 2-hand, Touch, Absolute), P2TR (Phone, 2-hand, Touch, Relative), P1RR (Phone, 1-hand, Rotate, Relative), W2TR (Watch, 2-hand, Touch, Relative), W1RR (Watch, 1-hand, Rotate, Relative), respectively.

Device	Absolute Touch-based (TA)	Relative Touch-based (TR)	Relative Rotation-based (RR)
Phone	P2TA	P2TR	P1RR
Watch		W2TR	W1RR

Table 3.1: Five representative pointing techniques with smartphone and smartwatch in three different pointing metaphors.

In the remainder of this section, we provide implementation details for the relative touch (phone and watch), absolute touch (phone only), and relative rotation (phone and watch) techniques respectively. We then describe the selection mechanism (clicking) for phone and watch-based techniques.

3.2.1 Relative Touch-based techniques (P2TR, W2TR)

The relative touch-based techniques (P2TR, W2TR) are operated via touch events on the display of devices, using the full display of the smartphone (landscape orientation) or the full display of the smartwatch as a trackpad. Cursor movement is relative to the finger's movement on the touchscreen and moves along the virtual wall. When the cursor hits another virtual wall at the intersection of two walls, it will change its position from the current wall to the next wall such that relative touch-based techniques on the 2D touchscreen can be applied in 3D space. We used a generalized logistic function [47] to define the CD gain between the move events on the touchscreen and cursor displacement on the virtual walls:

$$CD(v) = \frac{(CD_{Max} - CD_{Min})}{1 + e^{-\lambda \times (v - V_{inf})}} + CD_{Min}$$

CD_{Max} and CD_{Min} are the asymptotic maximum and minimum amplitudes of CD gains and λ is a parameter proportional to the slope of the function at $v = V_{inf}$ with V_{inf} a inflection value of the function.

The initial values of parameters were first generated by definitions from [47] and empirically optimized via pilots to control speed for viewport-based pointing for smartphones and smartwatches separately. The parameters were not changed during the study for individual participants nor for world-based because clutching costs in world-based pointing did not impact performance [48]; values are summarized in Table 3.2.

Device	CD_{Max}	CD_{Min}	λ	V_{inf}
Phone	20.8 mm/mm	0.0250 mm/mm	40 s/mm	0.053 mm/s
Watch	25.4 mm/mm	0.0125 mm/mm	88 s/mm	0.025 mm/s

Table 3.2: Logistic function parameters for relative touch-based techniques.

3.2.2 Absolute Touch-based techniques (P2TA)

The absolute touch-based techniques (P2TA) map a user's finger location on the touchscreen of a personal device to a fixed, consistent position of the virtual cursor on either the viewport or the virtual wall.

Absolute touch-based techniques enable users to position the virtual cursor easily on the mapped area. For VB, we mapped the smartphone's touchscreen to the viewport of the HMD, such that the range of actions was limited to the viewport. For WD, the 360° environment did not make sense in terms of mapping. Instead, we mapped the smartphone's touchscreen to the virtual wall that the camera of the HMD faced, such that the virtual cursor movement was constrained by the size of the corresponding wall. Users can switch to another virtual wall by rotating the HMD on their head and looking at that surface. The absolute mapping is then applied to cursor location on the corresponding surface.

3.2.3 Relative Rotation-based techniques (P1RR, W1RR)

We implemented fixed-origin raycasting for relative rotation-based techniques (P1RR and W1RR in Table 3.1), where the origin of the ray is set to be a fixed point, located at the camera of the HMD. Unlike a classical version of raycasting [38] with a moving origin, fixed-origin raycasting eliminates the need to reliably track the personal device in 3D space [54] and uses only the devices' built-in sensors. The 9-axis inertial measurement units (IMUs) of smart devices, including the accelerometer, the geomagnetic field sensor, and the gyroscope, enables monitoring the orientation of the device. Relative rotation-based techniques map changes in device orientation on the Yaw axis ($\Delta\alpha$) and Pitch axis ($\Delta\beta$) (in degrees) to the ray orientation ($\overrightarrow{Ray}$) in three axes, illustrated in Figure 3.3 using the following mappings:

$$Q_{Yaw} = F_{AA2Q}(\Delta\alpha, \overrightarrow{oy})$$

$$Q_{Pitch} = \begin{cases} F_{AA2Q}(\Delta\beta, \overrightarrow{oz}), & \text{if } \overrightarrow{Ray} \text{ points at Left/Right wall} \\ F_{AA2Q}(\Delta\beta, \overrightarrow{ox}), & \text{otherwise} \end{cases}$$

$$\overrightarrow{Ray} = Q_{Pitch} \times Q_{Yaw} \times \overrightarrow{Ray}$$

F_{AA2Q} is a transformation function that converts Axis-Angle to Quaternion, and $\overrightarrow{oz}$, $\overrightarrow{oy}$,$\overrightarrow{ox}$, are defined as unit vectors along each local axis from the origin of the ray respectively. The position of the cursor is the intersection point when the ray passes through the virtual wall.

We did not apply any cursor acceleration mechanism because we were not mapping device orientation to its displacement on the display [61, 31]. Therefore, the CD gain is set to 1:1, with exact correspondence between the ray and the devices' orientation.

In our pilot study, we found that even the weak magnetic field in the experimental environment was able to cause unstable sensor readings, which directly resulted in jittering of the ray and unstable cursor movement. As [33] suggested, calibration to a specific single environment was commonly required to reduce electromagnetic interference for magnetic systems. Therefore, in our implementation, instead of using Android SDK's Rotation Vector to measure the orientation of devices [31, 32], which is a result of sensor fusion of the gyroscope, accelerometer and magnetometer sensors, we used the Game Rotation Vector, which is identical to the Rotation Vector but not impacted by magnetic field changes. We also applied virtual sensor fusion [51] to reduce integrated drift during the study and improve accuracy and stability.

3.2.4 Postures of Pointing Techniques

Figure 3.2 illustrates postures and valid touch areas when performing the mentioned pointing techniques using smartphones and smartwatches above: (a) Touch-based techniques (P2TR & P2TA) use landscape orientation for two-hand use. (b) Relative rotation-based techniques (P1RR) have the touch surface positioned in the lower-half to accommodate one-hand use. (c) Touch-based techniques (W2TR) use a round surface for two-hand use. (d) Relative rotation-based techniques (W1RR) use a wrist-worn wearing method for one-hand use.

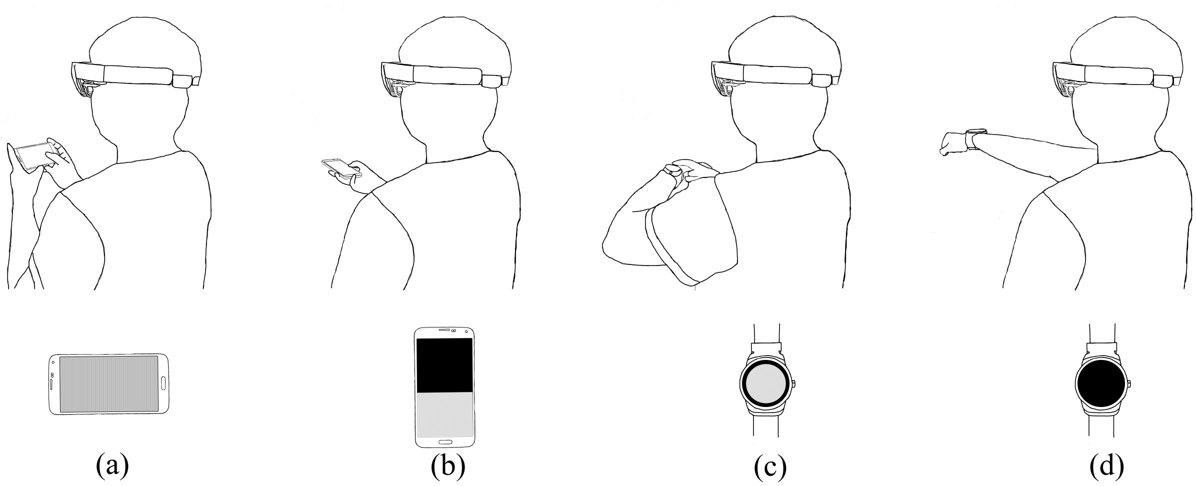

Figure 3.2: Postures for manipulating cursor using smart devices with valid touch area coloured as grey.

3.2.5 Clicking for Selection

For touch-based techniques, 'clicking', the selection action, is triggered by tapping the touchscreen; if taps are short enough in duration ($T_{up} - T_{down} \leq 500\,\text{ms}$), we map them to a selection action. For P1RR, the tapping area is 64 mm × 60 mm large and fully covers the lower part of smartphone to allow users to smoothly interact with the touchscreen and avoid missed clicks [9].

To select a target using W1RR, we leveraged Katsuragawa et al.'s wrist rotation technique [31]. Users rotate their wrist outwards over 30° and back in 1 s. However, the watch may detect horizontal movements with wrist rotation when triggering a selection, which can cause the cursor to be displaced during the clicking movement and introducing a selection error, a phenomenon known as Heisenberg Effect [12]. To handle this issue, we applied Katsuragawa et al.'s book-keeping design for clicking correction [31]: when the speed of wrist rotation is less than 10 °/s, the wrist rotation state is defined as neural and it is defined as left/right depending on its rotation direction otherwise. When the wrist rotation state changes, the direction of the ray is stored in memory such that when a click event is triggered by the user, the ray is mapped to its saved location, providing selection occurs within a predefined timeout of 1 s.

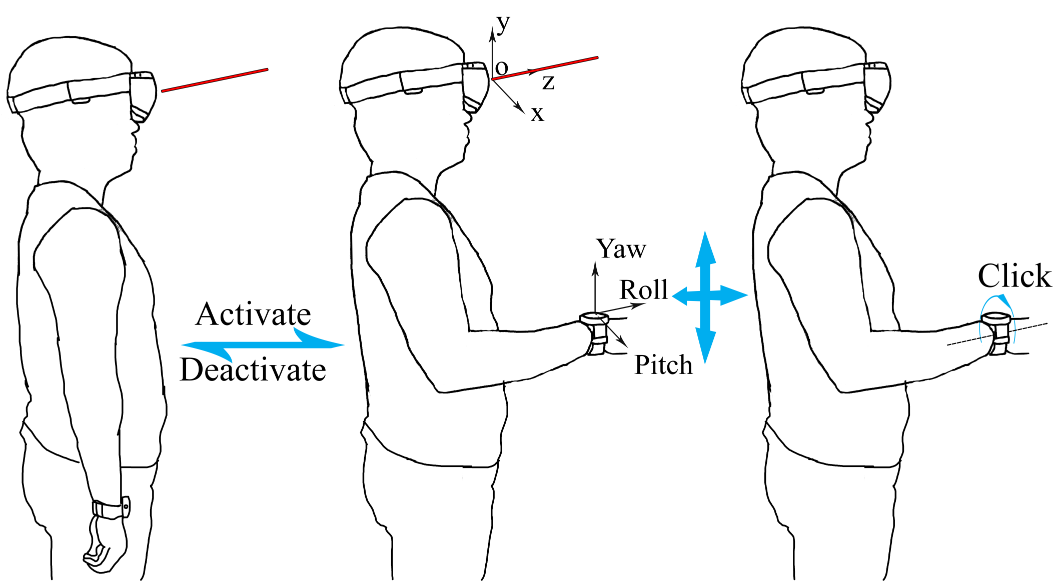

Figure 3.3: Manipulate a cursor and trigger selection using W1RR.

Figure 3.3 describes the procedures to manipulate a cursor and trigger selection using a smartwatch. Users raise and put down their dominated forearm to activate and deactivate cursor manipulation, and then users must sweep the entire forearm to cause changes orientation of the smartwatch. To trigger a click action, a 30° wrist flick outwards and back in 1 s is performed.

3.2.6 Methodology Summary

To conclude the methodology section, We summarize pointing conditions generated by combinations of pointing paradigms and pointing techniques and matched with corresponding selection mechanisms as shown in Figure 3.4.

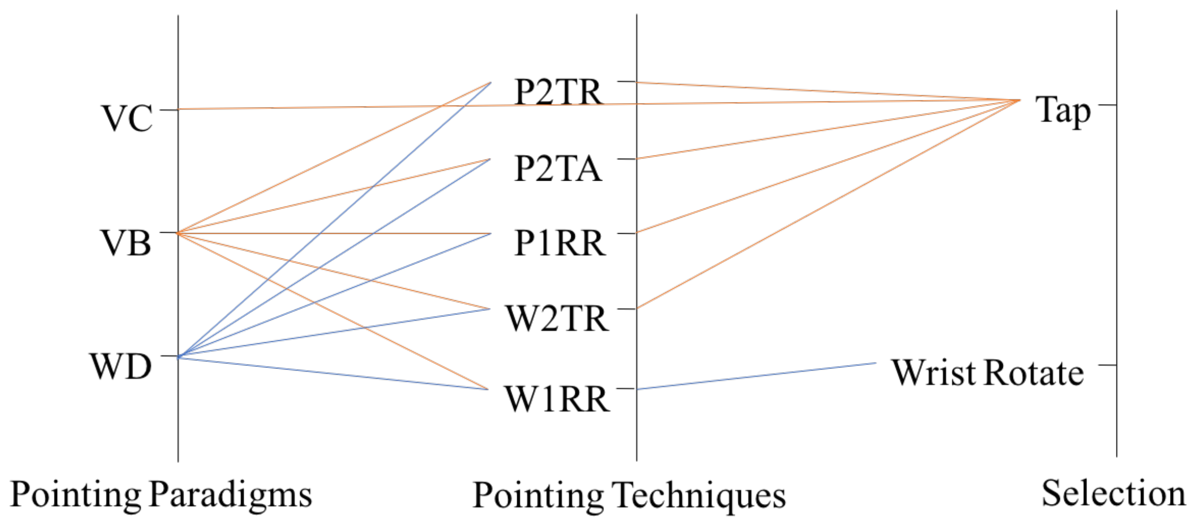

Figure 3.4: 11 pointing conditions generated by combinations of 3 pointing paradigms and 2 pointing techniques.

Chapter 4

General Experiment Setup

Combining our experimental factors – viewport- versus world-based pointing and five pointing techniques – yields ten experimental conditions. Adding Viewport-Centre pointing as a control condition yields 11 conditions, summarized in Figure 3.4. In early piloting of the experiment with 11 interaction conditions, we found that, in order to obtain a sufficient number of selection actions, the experiment using both smartphone and smartwatch techniques took almost two hours to complete.

To address this, we separated our experiment into two studies: one with smartphone and one with smartwatch. For consistency across experimental conditions, the apparatus, implementation, environment and task were identical in both conditions. In the remainder of this section, we will describe the common experimental setup.

4.1 AR Apparatus and Implementation

We used Microsoft Hololens (1st generation) [44] as the HMD. The augmented reality system was implemented in the Unity 2017.4.27f1 engine [71] with the HoloToolkit [43] for spatial mapping and understanding. Pointing techniques with phone and watch were implemented in Android and the communication between the AR system and phone/watch was achieved via a TCP server written in C#.

4.2 Environment

As shown in Figure 4.1, we used fabric walls to create a cubic room such that we were able to use spatial mapping to model the environment and generate four virtual walls. The room measured 230 cm × 230 cm × 250 cm in both real and virtual space.

In the pilot study, participants had problems selecting targets using WDW1RR, the world- and watch-based, one-handed, rotation-relative technique when targets were generated close to the upper and lower edges of virtual walls. Therefore, the height from the highest target to the lowest target was set to 210 cm. To visually present the valid activity range of the virtual space, we used two cyan spheres to indicate the corners of the space, with one located at the intersection of the *Front* and *Left* walls and another located at the intersection of the *Back* and *Right* walls. Their positions indicate the highest and lowest position of targets in the AR space respectively. There was an unanticipated benefit of these spheres: when participants occasionally experienced Hololens tracking issues, they were asked to wait until the Hololens recovered. On recovery, the camera position was shifted, but virtual objects' relative positions did not change. Cyan spheres described above were used to quickly recenter the camera so participants could continue the experiment.

The experiment was conducted as a seated experience. Participants sat on a swivel chair to allow convenient rotation and avoid the risk of falling that can arise in a standing experience. The chair was located at the center of the space initially but participants were able to move around in the space.

For the duration of the experiment, the Hololens was connected to a laptop via Holographic Remoting Player, and both the smartphone and smartwatch were connected to the same laptop via a TCP server. These two connections were established through a private WI-FI network. To guarantee stable data transmission for smooth control, internet access was disabled.

Participants were asked to use two hands to perform the touch-based techniques and to use their dominant hand to hold the smartphone or wear the smartwatch when using rotation-based techniques. Figure 3.2 & 3.4 summarize the touch surfaces, available device orientation, and selection mechanisms for interaction. These studies took place in a closed lab on our university campus.

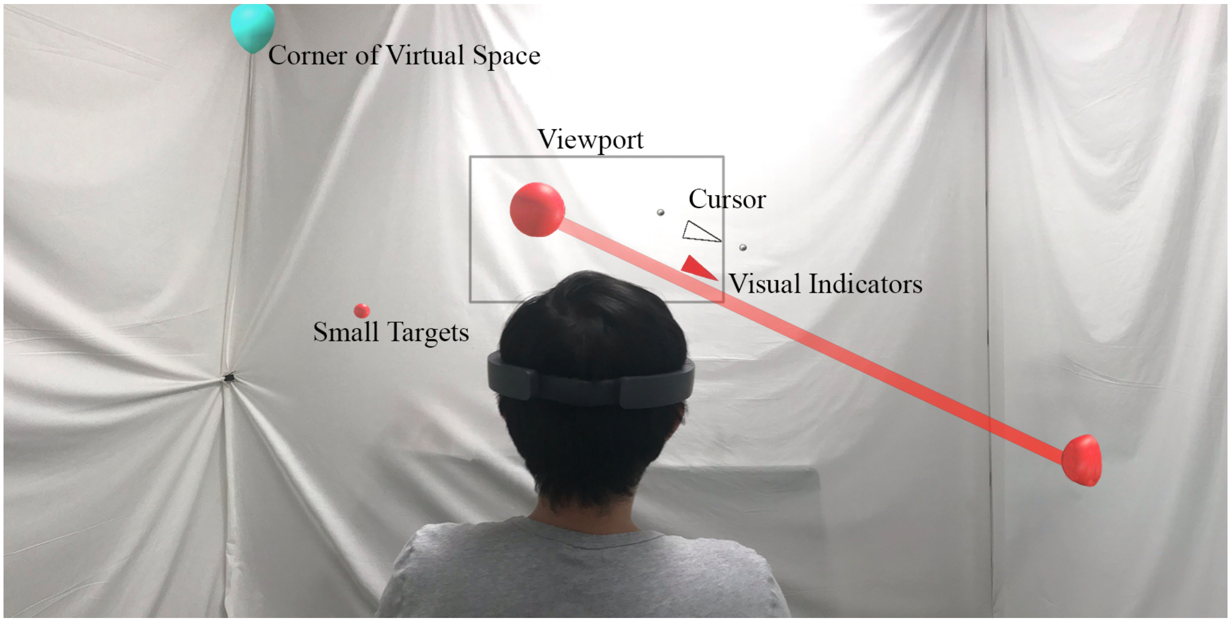

Figure 4.1: The experiment setup and real and virtual environments a participant can see through the Hololens.

4.3 Common Protocol

To efficiently utilize the interaction space, spheres (targets) are generated and distributed on the four walls in the AR environment. A repeated-measure within-subject design was used. The independent variables (IVs) were *Width* (2.5 cm, 10.0 cm) and *Distance* (60.0 cm, 90.0 cm, 120.0 cm (on the same wall), 200.0 cm (from the wall the current target is on to its neighboring wall in clockwise and counter-clockwise direction), 237.7 cm (from the wall the current target is on to its opposite wall: *Front-Back* and *Left-Right*)), *Block* (1-3) and *Pointing Condition* (which varied for Smartphone versus Smartwatch studies). *Pointing Condition* can be refactored into two factors, i.e. *Pointing Paradigm* (viewport versus world-based) and *Pointing Technique* for further analysis. The index of difficulties (ID) of the experimental task ranged from 2.81 to 6.59 bits. The order of Pointing Condition was counterbalanced across participants using a Latin square [76].

For each trial, participants had to select a target of width W and located at a distance D from the previous position of the cursor. To select a target, participants positioned the cursor over the target and select it using the designated interaction technique. The cursor was bright white and all targets were dark red. When participants positioned the cursor over a target, the target turned yellow, which visually informed participants that they had acquired the target. The experimental system moved to the next trial only when the target was correctly selected. If the target selection was inaccurate, the target did not disappear. Once correctly selected, the current target vanished and the next target was generated on one of the four walls. Only one target was displayed at a time.

To help participants quickly find targets under different viewing conditions, a red line connected the vanished target to the next target in AR space. In our pilot study, we found that when D was 237.7 cm (i.e. opposite wall, *behind* participants) and a red line passed through the camera, participants would lose track of the line. Therefore, we also used a red arrow to indicate the direction of a target when it was outside of the viewport. In WD, a white arrow was displayed to indicate the direction of the cursor when the cursor was outside of the viewport.

To more accurately simulate pointing in the real world, in each block the target width was randomly generated from within the study target widths. Given a target width, a "dummy" target was randomly generated on the wall that the user was looking at to control the initial position of the cursor and use its selection time as a referent for measuring the selection time of the next target. Then, targets with each *Distance* were generated twice. Considering this, participants performed $(2\ (\textit{Width}) \times 7\ (\textit{Distance}) + 1\ (\textit{Dummy})) \times 2 = 30$ targeting tasks per block. A block design was used to allow a block analysis to reveal significant learning effects. We asked each participant to perform three blocks per pointing condition.

4.4 Procedure

Before the study started, the researcher removed the stored virtual objects (holograms) in the Hololens from the previous study sessions and scanned the environment to stabilize Hololens tracking. Participants were welcomed to the environment, consent was obtained, and they were fitted with the Hololens.

Based on counterbalancing, participants began with the first interaction condition, performed 90 pointing tasks with that condition, and then removed the HoloLens. Participants took a one-minute break and completed a raw NASA TLX Questionnaire [26] to assess the pointing condition. Participants then went on to the next pointing condition. The study lasted for 60 minutes and participants received $15 for remuneration.

Chapter 5

Study 1: Smartphone

Our smartphone experiment included seven *Pointing Conditions*: Viewpoint Centre (VC), our control condition, plus three *Pointing Techniques* (Phone-2hand-Touch-Relative; Phone-2-hand-Touch-Absolute; and Phone-1hand-Rotate-Relative) for each of Viewport-Based and World-Based *Pointing Paradigms*. These are abreviated: VBP2TR, VBPT2TA, VBP1RR, WDPT2TR, WDP2TA, WDP1RR.

5.1 Participants and Apparatus

We recruited 12 participants aged from 22 to 28 ($\mu = 24.33$ and $\theta = 2.02$), two female, two left-handed, one ambidextrous. Three participants had AR/VR experience and all were familiar with touch-based devices. Participants were drawn from students from the university campus through a recruitment email. Participants used a Samsung Galaxy S5 (rectangular display of 113.50 mm × 64.00 mm and resolution of 1920 px × 1080 px, for a pixel density of 169.06 $^{\text{px}}/_{\text{cm}}$) as the input device.

5.2 Results

In the following results, we refer to completion time as CT and error rate and offset distance (in cm) of selection per block as $\%_{Err}$ and OD respectively. A mis-selection is counted when the cursor hovers over a target but, when a click event is then triggered, the cursor is mapped to a location outside the target.

We conducted a repeated-measures ANOVA ($\alpha = 0.05$) for completion time and accuracy with two independent variables: *Block* and *Pointing Condition*. Therefore, 12 participants $\times$ 7 *Pointing Condition* $\times$ 3 *Block* $\times$ 28 targets = 7056 trials. Because there was no pointing technique associated with VC, we aggregated a dataset excluding VC (6048 of 7056 trials, or 85.7%), and refactored *Pointing Condition* as *Pointing Paradigm* and *Pointing Technique*. The post-hoc tests were conducted using pairwise t-test with Bonferroni corrections when significant effects were found. Effect sizes are reported as partial eta squared (η_p^2). We validated all assumptions of the analysis of variance and completed the data analysis and visualization in R.

Order Effects

We found no significant effect of the ordering of *Pointing Condition* on either CT, $\%_{Err}$ or OD. We found a significant effect of *Block* on CT ($F_{2,22} = 16.40, p < 0.001, \eta_p^2 = 0.05$) but no significant effect of *Block* on $\%_{Err}$ or OD. Pairwise t-test reported that trials in Block 1 (mean 5997.69 ms) took significantly longer than Block 2 ($5878.75\,\mathrm{ms}, p < 0.001$) and Block 3 ($5674.46\,\mathrm{ms}, p < 0.001$). While effect size is small, we remove Block 1 in our analysis due to its statistically significantly slower movement time, leaving 4704 of 7056 trials.

5.2.1 Completion Time

We used the median, rather than the mean, to compensate for the non-normal distribution of CT. We found a significant effect of *Pointing Condition* on CT ($F_{6,66} = 17.18, p < 0.001, \eta_p^2 = 0.57$). The pairwise comparisons between pointing conditions are shown in Figure 5.1. The statistical significances evaluated by pairwise t-test are marked with stars ($** = p < 0.01$ and $* = p < 0.05$).

As a control pointing condition, VC (median 4109.00 ms) was significantly faster than other pointing conditions. By refactoring Pointing Condition into *Pointing Paradigm* and *Pointing Technique*, we found significant effects of *Pointing Paradigm* and *Pointing Technique* on CT ($F_{1,11} = 32.18, p < 0.001, \eta_p^2 = 0.28$ and $F_{2,22} = 5.01, p < 0.05, \eta_p^2 = 0.11$ respectively). Pointing techniques with VB, VBP2TR (4665.50 ms), VBP2TA (4998.70 ms) and VBP1RR (4948 ms) were all significantly faster than those with WD, WDP2TR (5677.00 ms), WDP2TA (6082.30 ms) and WDP1RR (5244.00 ms). We also found that absolute touch-based techniques were significantly slower than relative touch-based and rotation-based pointing techniques ($p < 0.001$ for both). We did not find a significant interaction effect between *Pointing Paradigm* and *Pointing Technique*.

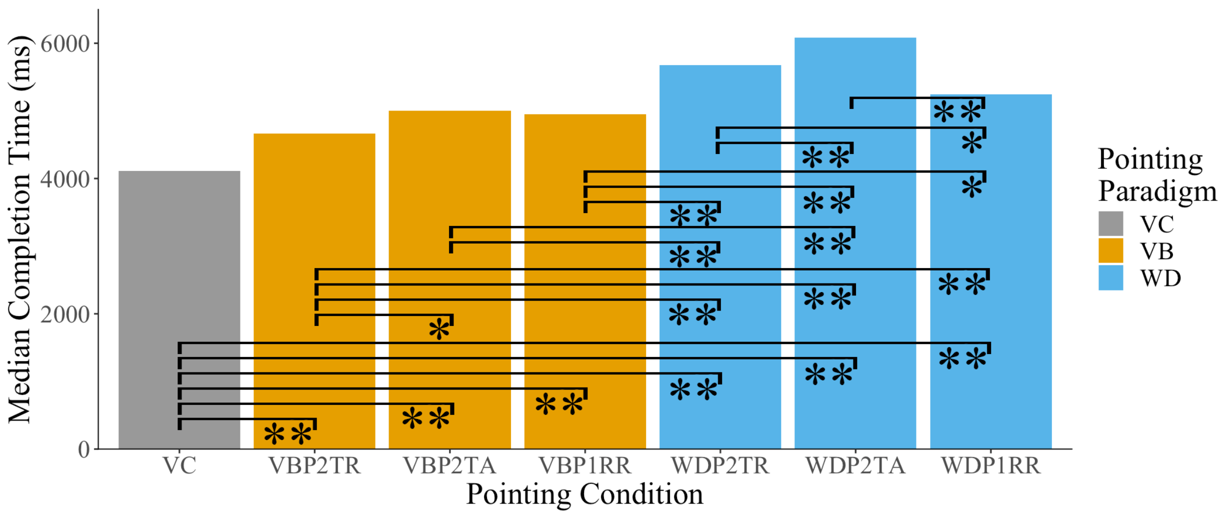

Figure 5.1: Median completion time for pointing conditions.

5.2.2 Error rate & Offset Distance

Because participants had to successfully complete a target selection before moving to the next trial, we report both ($\%_{Err}$) and Offset Distance (OD). We found a significant effect of *Pointing Condition* on $\%_{Err}$ ($F_{6,66} = 15.02, p < 0.001, \eta_p^2 = 0.50$) and on OD ($F_{6,66} = 8.17, p < 0.001, \eta_p^2 = 0.33$). The pairwise comparisons between pointing conditions were shown in Figure 5.2. The corresponding error bars are visualized to indicate standard deviation. The statistical significances evaluated by pairwise t-test are marked with stars ($** = p < 0.01$ and $* = p < 0.05$).

While VC was faster than all other pointing conditions, it had much higher error rate (mean 27.53%) and relatively high offset distance (mean 1.15 cm). We found significant effects of *Pointing Paradigm* and *Pointing Technique* on $\%_{Err}$ ($F_{1,11} = 23.25, p < 0.001, \eta_p^2 = 0.15$ and $F_{2,22} = 17.27, p < 0.001, \eta_p^2 = 0.34$ respectively) and OD ($F_{1,11} = 13.62, p < 0.005, \eta_p^2 = 0.10$ and $F_{2,22} = 8.26, p < 0.005, \eta_p^2 = 0.17$ respectively). Generally, pointing techniques with the world-based metaphor – WDP2TR (10.86%, 0.54 cm), WDP2TA (9.38%, 0.91 cm) and WDP1RR (22.78%, 0.57 cm) – caused fewer erroneous clicks and smaller offset distance than those with viewport-based metaphor – VBP2TR (12.05%, 0.53 cm), VBP2TA (25.89%, 1.89 cm) and VBP1RR(24.70%, 0.86 cm). We also found a significant interaction effect between *Pointing Paradigm* and *Pointing Technique* on $\%_{Err}$ ($F_{2,22} = 9.44, p < 0.005, \eta_p^2 = 0.19$) and OD ($F_{2,22} = 5.74, p < 0.01, \eta_p^2 = 0.09$). This appears to be because P2TA caused fewer erroneous clicks and smaller offset distance in

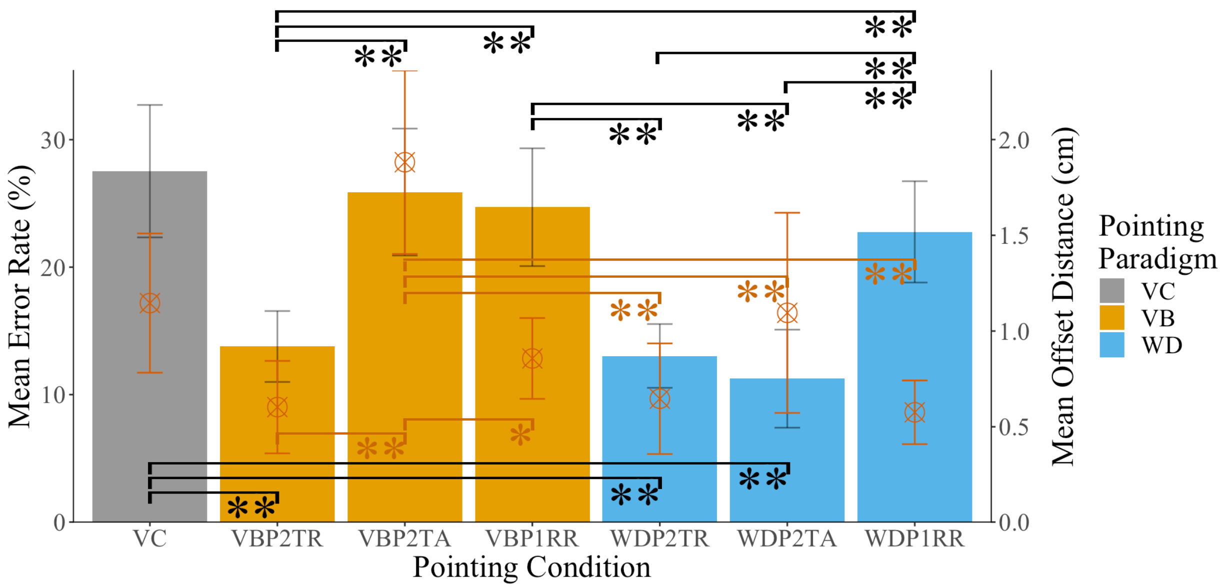

Figure 5.2: Mean error rate (bar) and offset distance (cross circle) for pointing conditions.

WD, compared with its performance in VB while the error rate and offset distance of other pointing techniques remained relatively constant within VB and WD.

5.2.3 NASA TLX

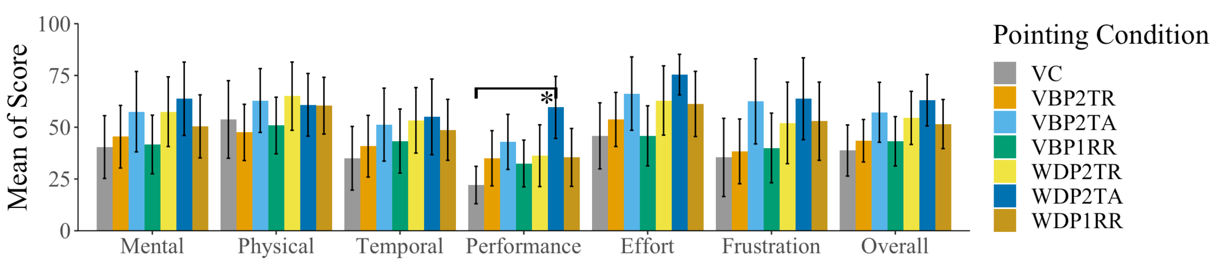

Figure 5.3: The mean responses for the attributes of NASA TLX questionnaire.

Results in Figure 5.3 showed differences for perceived task load between pointing techniques. The standard error bar is visualized. The statistical significances evaluated by Wilcoxon rank sum test are marked with stars ($** = p < 0.01$ and $* = p < 0.05$).

A Friedman test showed significant effect of *Pointing Condition* on attributes except *Physical*: $\chi^2_{Mental}(6) = 21.38, p < 0.01$, $\chi^2_{Temporal}(6) = 17.35, p < 0.01$, $\chi^2_{Performance}(6) = 25.17, p < 0.001$, $\chi^2_{Effort}(6) = 20.31, p < 0.005$, $\chi^2_{Frustration}(6) = 22.78, p < 0.001$, $\chi^2_{Overall}(6) = 22.42, p < 0.005$. However, post-hoc tests between pointing conditions did not highlight any statistically significant differences between conditions expect for *Performance*, and then only for World-2Hand-Touch-Absolute versus the control condition.

To explore this question more fully, recall that *Pointing Condition* comprises two factors: *Pointing Paradigm* (VB vs WD) and *Pointing Technique* (P2TR, P2TA, P1RR). Analyzing NASA TLX results, we found a significant effect of *Pointing Paradigm* on attributes except *Physical*, *Performance* and *Frustration*: $\chi^2_{Mental}(1) = 8.33, p < 0.005$, $\chi^2_{Temporal}(1) = 4.45, p < 0.05$, $\chi^2_{Effort}(1) = 4.45, p < 0.05$, $\chi^2_{Overall}(1) = 8.33, p < 0.005$. We found a significant effect of *Pointing Technique* only on attributes *Frustration* and *Overall*: $\chi^2_{Frustration}(2) = 10.50, p < 0.01$, $\chi^2_{Overall}(2) = 6.43, p < 0.05$.

Examining the graphical results and interpreting them in light of statistical analyses, these results argue that the viewport-based pointing conditions have lower task load than the world-based pointing conditions, an effect that seems to hold regardless of technique. Interestingly, the perceived *Mental*, *Physical*, *Effort* for VC was rated approximately equal or higher than VBP2TR or VBP1RR, which requires more hand movement overall.

5.2.4 Fitts' Law

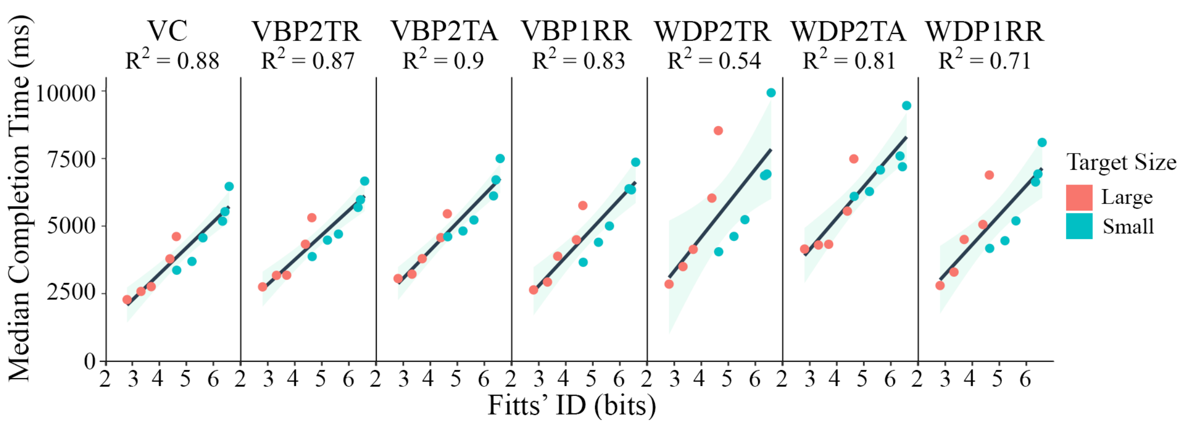

Figure 5.4: Median completion time as a function of Fitts' ID per pointing condition, with corresponding R^2 and confidence interval.

We aggregated the median CT for each Fitts' Index of Difficulty (ID) and each pointing condition. The Fitts' ID ranged from 2.81 to 6.59 bits and the aggregated time of all pointing conditions correlate with Fitts' ID positively (Figure 5.4) [19].

Except WDP2TR ($R^2 = 0.54$) and WDP1RR ($R^2 = 0.71$), pointing conditions have $R^2 \geq 0.8$. Analyzing WDP2TR interaction, from field notes we found that, when a target is generated outside of the FOV (and particularly behind the participant), the continuous moving of the finger on the screen without knowing the position of the cursor will cause the cursor to exceed the target position multiple times such that extra physical effort is needed to move the cursor back. VBP1RR and WDP1RR with the smartphone have a similar issue, but it is caused by difficulties in holding the phone and in controlling cursor stably with one hand. Essentially, the jittering of the cursor around targets adds slightly more physical effort to these interactions.

Chapter 6

Study 2: Smartwatch

The Smartwatch study differed from the Smartphone study only in *Pointing Condition*. *Pointing condition* had 5 levels: Viewport Centre, VC, our control condition, plus two *Pointing Techniques* (Watch-2hand-Touch-Relative; and Watch-1hand-RotateRelative) for two *Pointing Paradigms* (Viewport-Based vs World-Based), i.e. VBW2TR, VBW1RR, WDW2TR, WDW1RR.

6.1 Participants and Apparatus

We recruited 12 new participants aged from 21 to 31 ($\mu = 24.50$ and $\theta = 2.47$), one female, all right-handed, two with AR/VR experience. All were familiar with touch-based devices. Only one owned a smartwatch (Apple Watch); the others had owned a normal wristwatch. Participation in Study 1 was an exclusion criteria to avoid introducing biases in the smartwatch study. Participants used an LG G Watch R (round display of 33.44 mm and resolution of 320 px × 320 px, for a pixel density of 97 px/cm) as the input device.

6.2 Results

We conducted a repeated-measures ANOVA ($\alpha = 0.05$) for completion time and accuracy with same independent variables as in Study 1 and followed a similar analysis. There were 5040 trials in total and the aggregated dataset excluding VC contained 4032 of 5040 trials (80%).

6.2.1 Order Effects

We found no significant effect of the ordering of *Pointing Condition* on either CT, $\%_{Err}$ or OD. We found significant effects of *Block* on CT ($F_{2,22} = 25.37, p < 0.001, \eta_p^2 = 0.03$) but no significant effects of it on $\%_{Err}$ or OD. Pairwise t-test reported that trials in Block 1 (5777.74 ms) took significantly longer than Block 2 ($5558.81\,\text{ms}, p < 0.05$) and Block 3 ($5405.80\,\text{ms}, p < 0.001$). While effect size for Block was small, due to the statistically significant effect of block, we removed Block 1 in our analysis, leaving 3360 of 5040 trials.

6.2.2 Completion Time

We found a significant effect of *Pointing Condition* on CT ($F_{4,44} = 17.38, p < 0.001, \eta_p^2 = 0.57$). The pairwise comparisons between pointing conditions are shown in Figure 6.1. The statistical significances evaluated by pairwise t-test are marked with stars ($** = p < 0.01$ and $* = p < 0.05$).

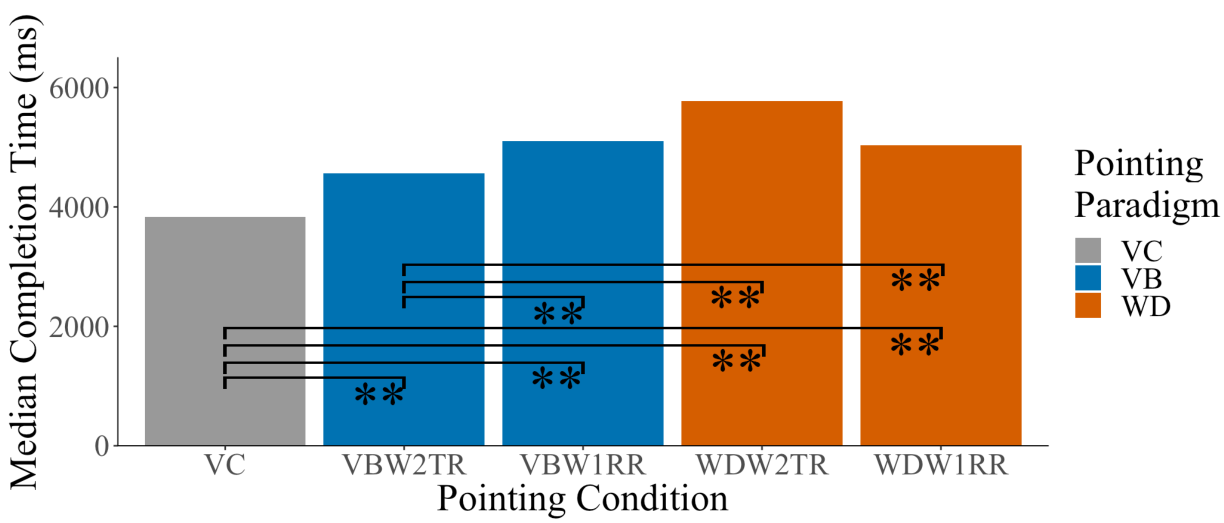

Figure 6.1: Median completion time for pointing conditions.

Similar to Study 1, VC was significantly faster (3830.0 ms) than other techniques when using smartwatch for input. We only found a significant effect of *Pointing Paradigm* ($F_{1,11} = 6.05, p < 0.05, \eta_p^2 = 0.08$) on CT. Similar to the results in Study 1, viewport-based pointing condition, VBW2TR (4559.0 ms) was faster than world-based pointing condition, WDW2TR (5769.5 ms) and VBW1RR (5105.0 ms) was approximately equal to WDW1RR (5031.0 ms).

6.2.3 Error Rate & Offset Distance

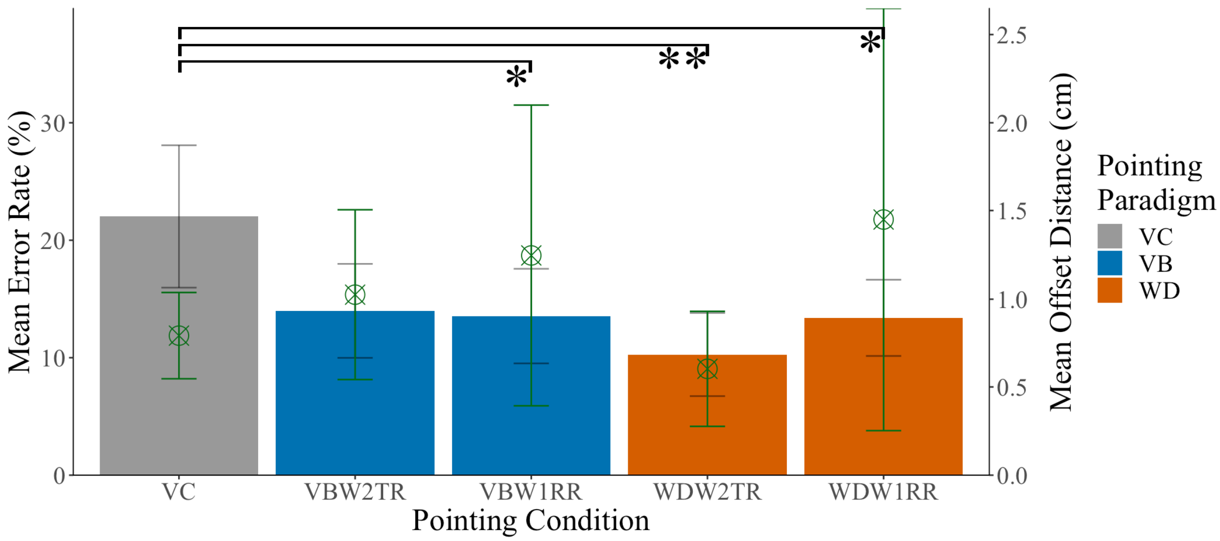

Figure 6.2: Mean error rate (bar) and offset distance (cross circle) for pointing conditions.

Again considering both Error rate, $\%_{Err}$, and Offset Distance, OD, we only found a significant effect of *Pointing Condition* on $\%_{Err}$ ($F_{4,44} = 6.44, p < 0.001, \eta_p^2 = 0.14$) but not on OD. The pairwise comparisons between pointing conditions are shown in Figure 6.2. The corresponding error bars are visualized to indicate standard deviation. The statistical significances evaluated by pairwise t-test are marked with stars ($** = p < 0.01$ and $* = p < 0.05$).

The result was similar to that in Study 1 with VC having the highest error rate (22.02%) but relatively low OD (0.79 cm). We found no significant effect of *Pointing Paradigm* or *Pointing Technique* on $\%_{Err}$ or OD.

6.2.4 NASA TLX

Results in Figure 6.3 showed differences for perceived task load between pointing techniques. The standard error bar is visualized. The statistical significances evaluated by

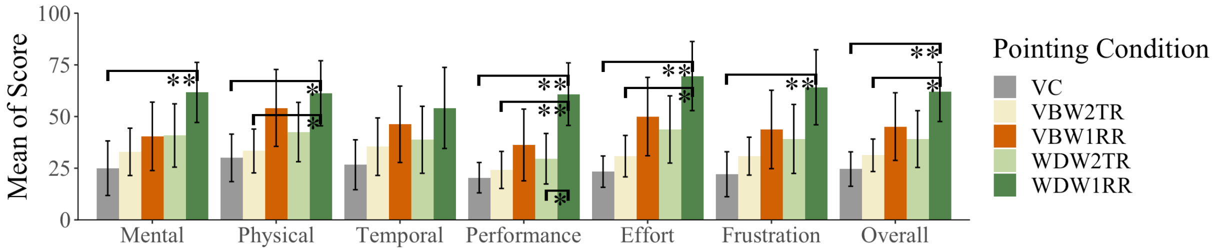

Figure 6.3: The mean responses for the attributes of NASA TLX questionnaire.

Wilcoxon rank sum test are marked with stars ($** = p < 0.01$ and $* = p < 0.05$).

The Friedman test results showed significant effect of *Pointing Condition* on all attributes: $\chi^2_{Mental}(4) = 18.82, p < 0.001$, $\chi^2_{Physical}(4) = 17.39, p < 0.005$, $\chi^2_{Temporal}(4) = 17.89, p < 0.005$, $\chi^2_{Performance}(4) = 22.79, p < 0.001$, $\chi^2_{Effort}(4) = 22.28, p < 0.001$, $\chi^2_{Frustration}(4) = 21.37, p < 0.001$, $\chi^2_{Overall}(4) = 23.98, p < 0.001$.

The significant effects of *Pointing Paradigm* on attributes were as follows: $\chi^2_{Mental}(1) = 5.33, p < 0.05$, $\chi^2_{Performance}(1) = 11.00, p < 0.001$, $\chi^2_{Effort}(1) = 5.33, p < 0.05$, $\chi^2_{Frustration}(1) = 5.33, p < 0.05$, $\chi^2_{Overall}(1) = 5.33, p < 0.05$. We also found a significant effect of *Pointing Technique* on attributes except for *Mental*: $\chi^2_{Physical}(1) = 5.33, p < 0.05$, $\chi^2_{Temporal}(1) = 7.36, p < 0.01$, $\chi^2_{Performance}(1) = 7.36, p < 0.01$, $\chi^2_{Effort}(1) = 5.33, p < 0.05$, $\chi^2_{Frustration}(1) = 7.36, p < 0.01$, $\chi^2_{Overall}(1) = 8.33, p < 0.005$.

Study 2 showed a consistent result as what Study 1 showed: viewport-based pointing have lower task load for all attributes than world-based pointing using the same pointing technique. However, task load comparisons for relative touch-based (TR) and relative rotation-based (RR) were different across studies: in Study 1, task loads for RR were rated equally or even lower than those for TR with the same pointing paradigm, while in Study 2, they were always rated higher with the same pointing paradigm, which is to be expected with the RR techniques with smartwatch: participants in Study 2 needed to raise their arms during the study, which demands additional effort.

6.2.5 Fitts' Law

Again we used median time values and aggregated for each Fitts' Index of Difficulty (ID) and each pointing condition. Fitts' ID ranged from 2.81 to 6.59 bits and the aggregated time of all pointing conditions correlate with Fitts' ID positively (Figure 6.4) [19]. As Figure 6.4 shows, except for WDW2TR ($R^2 = 0.67$), pointing conditions have $R^2 \geq 0.8$.

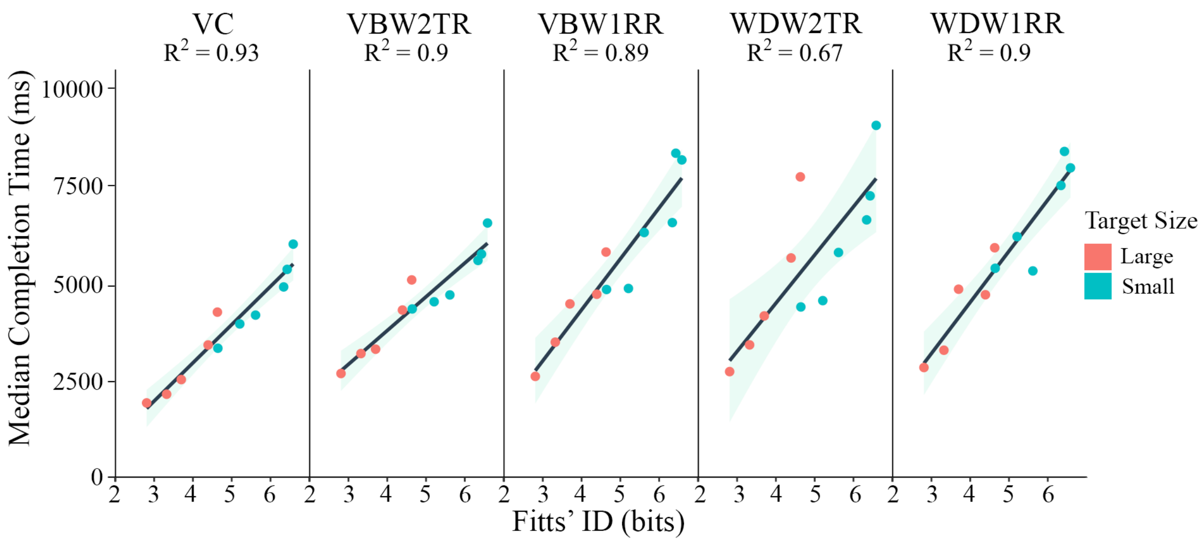

Figure 6.4: Median completion time as a function of Fitts' ID per pointing condition, with corresponding R^2 and confidence interval.

We observe a similar problem for touch relative input in world perspective. Because of the need to move the cursor a long distance, participants would lose track of the cursor, overshoot, and need to perform temporally costly corrections.

Chapter 7

Discussion

Overall, a straightforward interpretation of these results is as follows:

- Pointing techniques that leverage the viewport-based paradigm are faster, as accurate, and require less cognitive workload as measured on the NASA TLX for target selection.
- With the same pointing paradigm, relative touch-based techniques, i.e. touchpad-style input techniques, are faster and have lower cognitive load than other techniques. They cause fewer selection errors and lower offset distance from desired target.
- Relative rotation-based techniques have different performance across devices: they are faster and easier to control, but less accurate with lower offset distance for smartphones, while slower, harder to control but more accurate with larger offset distance for smartwatches.
- VC, gaze only pointing, was faster, but also had high errors and OD.

7.1 Design Implications

7.1.1 Viewport-based vs World-based

Our results show that viewport-based perspectives in selection improves speed without sacrificing accuracy for distant target acquisition. While the user searches for targets in viewport-based viewing, the cursor moves with head movement, which reduces the distance and time required to move the cursor compared with moving it via input devices in AR environments. Furthermore, with the cursor inside the view, participants felt less fatigued mentally and physically, due primarily to the cost of tracking cursor.

7.1.2 Choices of Pointing techniques and Device

For pointing technique, there appears to be a trade-off. VC while fastest, results in more errors and higher offset distance than other techniques, especially for small targets. Viewport- and relative touch-based techniques on phone and watch (VBP2TR and VBW2TR) are good alternatives if targets are on virtual surfaces [34]. If targets are generated in the air, the phone viewport-, relative rotation-based technique (VBP1RR) is a good alternative.

We observe one interesting effect: watch relative rotation-based techniques (VBW1RR or WDW1RR) (Figure 6.2) seem to cause fewer errors/higher offset than phone (Figure 5.2), contradicting Siddhpuria et al.'s [61] results on an external display. Examining this apparent contraction, we note two points. First, we did not map device orientation to displacement on the display, but, instead, directly applied a 1-to-1 mapping between device orientation and ray orientation. Second, we observed that when participants used VBW1RR or WDW1RR, they moved their forearm more carefully to keep track of the cursor in AR and rotated their wrist more deliberately to select targets, adding more mental and physical demand (see Figure 6.3). While previous studies [31, 32, 28, 55, 61] argue that it is feasible to use relative rotation-based techniques with a smartwatch to interact either with large displays or in virtual reality, our results show that using the forearm to manipulate cursor and wrist for selection may demand more physical effort than other techniques and cause higher offset distance and frustration (Figure 6.3 & 6.2).

7.2 Future Work

In future work, we would like to replicate this work on HMD-based VR displays and with specialized controllers. Such a study may further reveal how viewport- and world-based pointing paradigm affect users' performance and perception in virtual environments with virtual representations of controllers moving with head movement. We would also like to use our proposed pointing techniques to explore more complicated AR environments. More complex room shapes or additional furnishings may impact the feasibility of touch-based techniques due to irregular surfaces or rotation-based techniques due to environmental clutter.

Smart devices have an evolving set of on-board sensors that can capture orientation, vibration, and touch. New sensing technologies (e.g. [73]) promise to expand input on these personal devices into the space around the device. One final aim of future work is to fully use the increasing capabilities of these devices to perform tasks such as proximal object selection and virtual object manipulation.

Chapter 8

Conclusion

In this thesis, we explored a variety of smartphone-based and smartwatch-based pointing techniques for augmented reality interaction. These pointing techniques considered two factors: the pointing paradigm , i.e. viewport-based versus world-based; and pointing techniques, i.e. absolute or relative touch-based or rotation-based input. We found that, while equally accurate, viewport-based techniques provide faster and low-fatigue input. We also demonstrated, via high overall performance, that modern smart devices are effective input devices in augmented reality environments. As these personal devices expand in sensing capabilities, they provide an exciting, opportunistic platform to support rich input in head-mounted-display interactions.

References

[1] Johnny Accot and Shumin Zhai. Refining fitts' law models for bivariate pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, page 193–200, New York, NY, USA, 2003. Association for Computing Machinery.

[2] Ferran Argelaguet and Carlos Andujar. Efficient 3d pointing selection in cluttered virtual environments. *IEEE Computer Graphics and Applications*, 29(6):34–43, 2009.

[3] Ferran Argelaguet and Carlos Andujar. A survey of 3d object selection techniques for virtual environments. *Computers & Graphics*, 37(3):121 – 136, 2013.

[4] Teo Babic, Harald Reiterer, and Michael Haller. Pocket6: A 6dof controller based on a simple smartphone application. In *Proceedings of the Symposium on Spatial User Interaction*, SUI '18, pages 2–10, New York, NY, USA, 2018. ACM.

[5] Marc Baloup, Thomas Pietrzak, and Géry Casiez. Raycursor: A 3d pointing facilitation technique based on raycasting. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA, 2019. Association for Computing Machinery.

[6] Louis-Pierre Bergé, Emmanuel Dubois, and Mathieu Raynal. Design and evaluation of an "around the smartphone" technique for 3d manipulations on distant display. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, SUI '15, pages 69–78, New York, NY, USA, 2015. ACM.

[7] T. Bleeker, Gun Lee, and M. Billinghurst. Ego- and exocentric interaction for mobile ar conferencing. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 1–6, Oct 2013.

[8] Sebastian Boring, Dominikus Baur, Andreas Butz, Sean Gustafson, and Patrick Baudisch. Touch projector: Mobile interaction through video. In *Proceedings of the SIGCHI*

Conference on Human Factors in Computing Systems, CHI '10, pages 2287–2296, New York, NY, USA, 2010. ACM.

[9] Sebastian Boring, David Ledo, Xiang 'Anthony' Chen, Nicolai Marquardt, Anthony Tang, and Saul Greenberg. The fat thumb: Using the thumb's contact size for single-handed mobile interaction. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services*, MobileHCI '12, pages 39–48, New York, NY, USA, 2012. ACM.

[10] F. Bork, C. Schnelzer, U. Eck, and N. Navab. Towards efficient visual guidance in limited field-of-view head-mounted displays. *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2983–2992, Nov 2018.

[11] Doug A Bowman and Larry F Hodges. An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 35–ff, 1997.

[12] Doug A. Bowman, Chadwick A. Wingrave, J. M. Campbell, and V. Q. Ly. Using pinch gloves(tm) for both natural and abstract interaction techniques in virtual environments. 2001.

[13] Wolfgang Büschel, Annett Mitschick, Thomas Meyer, and Raimund Dachselt. Investigating smartphone-based pan and zoom in 3d data spaces in augmented reality. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '19, New York, NY, USA, 2019. Association for Computing Machinery.

[14] Jeffrey Cashion, Chadwick Wingrave, and Joseph J. Laviola. Dense and dynamic 3d selection for game-based virtual environments. *IEEE Transaction on Visualization and Computer Graphics*, pages 634–642, 2012.

[15] C. Christou, A. Tzanavari, K. Herakleous, and C. Poullis. Navigation in virtual reality: Comparison of gaze-directed and pointing motion control. In *2016 18th Mediterranean Electrotechnical Conference (MELECON)*, pages 1–6, April 2016.

[16] Nathan Cournia, John D. Smith, and Andrew T. Duchowski. Gaze- vs. hand-based pointing in virtual environments. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pages 772–773, New York, NY, USA, 2003. ACM.

[17] Barrett Ens, Fraser Anderson, Tovi Grossman, Michelle Annett, Pourang Irani, and George Fitzmaurice. Ivy: Exploring spatially situated visual programming for authoring and understanding intelligent environments. In *Proceedings of the 43rd Graphics Interface Conference*, GI '17, pages 156–162, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2017. Canadian Human-Computer Communications Society.

[18] P. M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47 6:381–91, 1954.

[19] Paul M Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology*, 47(6):381–391, June 1954.

[20] Clifton Forlines, Daniel Vogel, and Ravin Balakrishnan. Hybridpointing: Fluid switching between absolute and relative pointing with a direct input device. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, pages 211–220, New York, NY, USA, 2006. ACM.

[21] Tovi Grossman and Ravin Balakrishnan. Pointing at trivariate targets in 3d environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, page 447–454, New York, NY, USA, 2004. Association for Computing Machinery.

[22] Tovi Grossman and Ravin Balakrishnan. The bubble cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, page 281–290, New York, NY, USA, 2005. Association for Computing Machinery.

[23] Tovi Grossman and Ravin Balakrishnan. The design and evaluation of selection techniques for 3d volumetric displays. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, page 3–12, New York, NY, USA, 2006. Association for Computing Machinery.

[24] Uwe Gruenefeld, Dag Ennenga, Abdallah El Ali, Wilko Heuten, and Susanne Boll. Eyesee360: Designing a visualization technique for out-of-view objects in head-mounted augmented reality. In *Proceedings of the 5th Symposium on Spatial User Interaction*, SUI '17, pages 109–118, New York, NY, USA, 2017. ACM.

[25] Faizan Haque, Mathieu Nancel, and Daniel Vogel. Myopoint: Pointing and clicking using forearm mounted electromyography and inertial motion sensors. In *Proceedings*

of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15, pages 3653–3656, New York, NY, USA, 2015. ACM.

[26] Sandra G. Hart. Nasa-task load index (nasa-tlx); 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50(9):904–908, 2006.

[27] Juan David Hincapié-Ramos, Kasim Ozacar, Pourang P. Irani, and Yoshifumi Kitamura. Gyrowand: Imu-based raycasting for augmented reality head-mounted displays. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, SUI '15, pages 89–98, New York, NY, USA, 2015. ACM.

[28] Teresa Hirzle, Jan Rixen, Jan Gugenheimer, and Enrico Rukzio. Watchvr: Exploring the usage of a smartwatch for interaction in mobile virtual reality. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI EA '18, pages LBW634:1–LBW634:6, New York, NY, USA, 2018. ACM.

[29] HTC. Htc vive, 2019. https://www.vive.com/us/.

[30] Richard J Jagacinski, Daniel W Repperger, Sharon L Ward, and Martin S Moran. A test of fitts' law with moving targets. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 22(2):225–233, 1980.

[31] Keiko Katsuragawa, Krzysztof Pietroszek, James R. Wallace, and Edward Lank. Watchpoint: Freehand pointing with a smartwatch in a ubiquitous display environment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '16, pages 128–135, New York, NY, USA, 2016. ACM.

[32] Daniel Kharlamov, Brandon Woodard, Liudmila Tahai, and Krzysztof Pietroszek. Ticktockray: Smartwatch-based 3d pointing for smartphone-based virtual reality. In *Proceedings of the 22Nd ACM Conference on Virtual Reality Software and Technology*, VRST '16, pages 365–366, New York, NY, USA, 2016. ACM.

[33] Volodymyr V. Kindratenko. A survey of electromagnetic position tracker calibration techniques. *Virtual Real.*, 5(3):169–182, September 2000.

[34] Pascal Knierim, Valentin Schwind, Anna Maria Feit, Florian Nieuwenhuizen, and Niels Henze. Physical keyboards in virtual reality: Analysis of typing performance and effects of avatar hands. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 345:1–345:9, New York, NY, USA, 2018. ACM.

[35] R. Kopper, F. Bacim, and D. A. Bowman. Rapid and accurate 3d selection by progressive refinement. In *2011 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 67–74, 2011.

[36] Manu Kumar, Andreas Paepcke, and Terry Winograd. Eyepoint: Practical pointing and selection using gaze and keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, page 421–430, New York, NY, USA, 2007. Association for Computing Machinery.

[37] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst. Pinpointing: Precise head- and eye-based target selection for augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 81:1–81:14, New York, NY, USA, 2018. ACM.

[38] Jiandong Liang and Mark Green. Jdcad: A highly interactive 3d modeling system. *Computers & Graphics*, 18(4):499 – 506, 1994.

[39] I. Scott MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Hum.-Comput. Interact.*, 7(1):91–139, March 1992.

[40] I. Scott MacKenzie and William Buxton. Extending fitts' law to two-dimensional tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '92, page 219–226, New York, NY, USA, 1992. Association for Computing Machinery.

[41] Michael McGuffin and Ravin Balakrishnan. Acquisition of expanding targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, page 57–64, New York, NY, USA, 2002. Association for Computing Machinery.

[42] David E. Meyer, Richard A. Abrams, Sylvan Kornblum, Charles E. Wright, and J. E. Keith Smith. Optimality in human motor performance: ideal control of rapid aimed movements. *Psychological Review*, 95:340–370, 1988.

[43] Microsoft. Mixedrealitytoolkit-unity, Oct 2017. https://github.com/microsoft/MixedRealityToolkit-Unity/releases/tag/v1.2017.1.2.

[44] Microsoft. Microsoft hololens, 2019. https://www.microsoft.com/en-us/hololens.

[45] Peter Mohr, Markus Tatzgern, Tobias Langlotz, Andreas Lang, Dieter Schmalstieg, and Denis Kalkofen. Trackcap: Enabling smartphones for 3d interaction on mobile head-mounted displays. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, pages 585:1–585:11, New York, NY, USA, 2019. ACM.

[46] Atsuo Murata and Hirokazu Iwase. Extending fitts' law to a three-dimensional pointing task. *Human movement science*, 20(6):791–805, 2001.

[47] Mathieu Nancel, Olivier Chapuis, Emmanuel Pietriga, Xing-Dong Yang, Pourang P. Irani, and Michel Beaudouin-Lafon. High-precision pointing on large wall displays using small handheld devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 831–840, New York, NY, USA, 2013. ACM.

[48] Mathieu Nancel, Daniel Vogel, and Edward Lank. Clutching is not (necessarily) the enemy. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 4199–4202, New York, NY, USA, 2015. ACM.

[49] Oculus. Oculus rift s, 2019. https://www.oculus.com/rift-s/.

[50] Alex Olwal and S. Feiner. The flexible pointer: An interaction technique for selection in augmented and virtual reality. 2003.

[51] Alexander Pacha. *Sensor Fusion for Robust Outdoor Augmented Reality Tracking on Mobile Devices*. GRIN Verlag, USA, 2015.

[52] J. Karen Parker, Regan L. Mandryk, and Kori M. Inkpen. Tractorbeam: Seamless integration of local and remote pointing for tabletop displays. In *Proceedings of Graphics Interface 2005*, GI '05, pages 33–40, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.

[53] Julian Petford, Miguel A. Nacenta, and Carl Gutwin. Pointing all around you: Selection performance of mouse and ray-cast pointing in full-coverage displays. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 533:1–533:14, New York, NY, USA, 2018. ACM.

[54] Krzysztof Pietroszek, Anastasia Kuzminykh, James R. Wallace, and Edward Lank. Smartcasting: A discount 3d interaction technique for public displays. In *Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design*, OzCHI '14, pages 119–128, New York, NY, USA, 2014. ACM.

[55] Krzysztof Pietroszek, James R. Wallace, and Edward Lank. Tiltcasting: 3d interaction on large displays using a mobile device. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 57–62, New York, NY, USA, 2015. ACM.

[56] I. Poupyrev, T. Ichikawa, S. Weghorst, and M. Billinghurst. Egocentric object manipulation in virtual environments: Empirical evaluation of interaction techniques. *Computer Graphics Forum*, 17(3):41–52, 1998.

[57] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The go-go interaction technique: Non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, pages 79–80, New York, NY, USA, 1996. ACM.

[58] Ivan Poupyrev, Mark Billinghurst, Suzanne Weghorst, and Tadao Ichikawa. The go-go interaction technique: Non-linear mapping for direct manipulation in vr. In *Proceedings of the 9th Annual ACM Symposium on User Interface Software and Technology*, UIST '96, page 79–80, New York, NY, USA, 1996. Association for Computing Machinery.

[59] Hyocheol Ro, Seungho Chae, Inhwan Kim, Junghyun Byun, Yoonsik Yang, Yoonjung Park, and Tackdon Han. A dynamic depth-variable ray-casting interface for object manipulation in ar environments. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2873–2878. IEEE, 2017.

[60] I Scott MacKenzie and William Buxton. Prediction of pointing and dragging times in graphical user interfaces. *Interacting with Computers*, 6(2):213 – 227, 1994.

[61] Shaishav Siddhpuria, Sylvain Malacria, Mathieu Nancel, and Edward Lank. Pointing at a distance with everyday smart devices. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 173:1–173:11, New York, NY, USA, 2018. ACM.

[62] Frank Steinicke, Timo Ropinski, and Klaus Hinrichs. *OBJECT SELECTION IN VIRTUAL ENVIRONMENTS USING AN IMPROVED VIRTUAL POINTER METAPHOR*, pages 320–326. Springer Netherlands, Dordrecht, 2006.

[63] Frank Steinicke, Timo Ropinski, and Klaus H. Hinrichs. Object selection in virtual environments using an improved virtual pointer metaphor. In *ICCVG*, 2004.

[64] Sophie Stellmach and Raimund Dachselt. Look & touch: Gaze-supported target acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 2981–2990, New York, NY, USA, 2012. Association for Computing Machinery.

[65] Sophie Stellmach and Raimund Dachselt. Still looking: Investigating seamless gaze-supported selection, positioning, and manipulation of distant targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 285–294, New York, NY, USA, 2013. Association for Computing Machinery.

[66] Sophie Stellmach, Sebastian Stober, Andreas Nürnberger, and Raimund Dachselt. Designing gaze-supported multimodal interactions for the exploration of large image collections. In *Proceedings of the 1st Conference on Novel Gaze-Controlled Applications*, NGCA '11, New York, NY, USA, 2011. Association for Computing Machinery.

[67] Richard Stoakley, Matthew J. Conway, and Randy Pausch. Virtual reality on a wim: Interactive worlds in miniature. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, page 265–272, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[68] Vildan Tanriverdi and Robert J. K. Jacob. Interacting with eye movements in virtual environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 265–272, New York, NY, USA, 2000. ACM.

[69] Robert Teather and Wolfgang Stuerzlinger. Pointing at 3d targets in a stereo head-tracked virtual environment. pages 87 – 94, 04 2011.

[70] Robert J. Teather and Wolfgang Stuerzlinger. Pointing at 3d target projections with one-eyed and stereo cursors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 159–168, New York, NY, USA, 2013. Association for Computing Machinery.

[71] Unity. Unity 2017.4.27, May 2019. https://unity3d.com/unity/whats-new/2017.4.27.

[72] L. Vanacken, T. Grossman, and K. Coninx. Exploring the effects of environment density and target visibility on object selection in 3d virtual environments. In *2007 IEEE Symposium on 3D User Interfaces*, 2007.

[73] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 851–860, New York, NY, USA, 2016. ACM.

[74] Colin Ware and Ravin Balakrishnan. Reaching for objects in vr displays: Lag and frame rate. *ACM Trans. Comput.-Hum. Interact.*, 1(4):331–356, December 1994.

[75] Johann Wentzel, Greg d'Eon, and Daniel Vogel. Improving virtual reality ergonomics through reach-bounded non-linear input amplification. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–12, New York, NY, USA, 2020. Association for Computing Machinery.

[76] Ej Williams. Experimental designs balanced for the estimation of residual effects of treatments. 1949.

[77] Chadwick A. Wingrave and Doug A. Bowman. Baseline factors for raycasting selection. In *Proceedings of Virtual Reality International*, 2005.

[78] H. P. Wyss, R. Blach, and M. Bues. isith - intersection-based spatial interaction for two hands. In *3D User Interfaces (3DUI'06)*, pages 59–61, March 2006.

[79] Yukang Yan, Chun Yu, Xiaojuan Ma, Shuai Huang, Hasan Iqbal, and Yuanchun Shi. Eyes-free target acquisition in interaction space around the body for virtual reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 42:1–42:13, New York, NY, USA, 2018. ACM.

[80] Shumin Zhai, Carlos Morimoto, and Steven Ihde. Manual and gaze input cascaded (magic) pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, page 246–253, New York, NY, USA, 1999. Association for Computing Machinery.

[81] Fengyuan Zhu and Tovi Grossman. Bishare: Exploring bidirectional interactions between smartphones and head-mounted augmented reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–14, New York, NY, USA, 2020. Association for Computing Machinery.