



In-Depth Mouse: Integrating Desktop Mouse into Virtual Reality

Qian Zhou
Autodesk Research
Toronto, Canada
qian.zhou@autodesk.com

George Fitzmaurice
Autodesk Research
Toronto, Canada
george.fitzmaurice@autodesk.com

Fraser Anderson
Autodesk Research
Toronto, Canada
fraser.anderson@autodesk.com

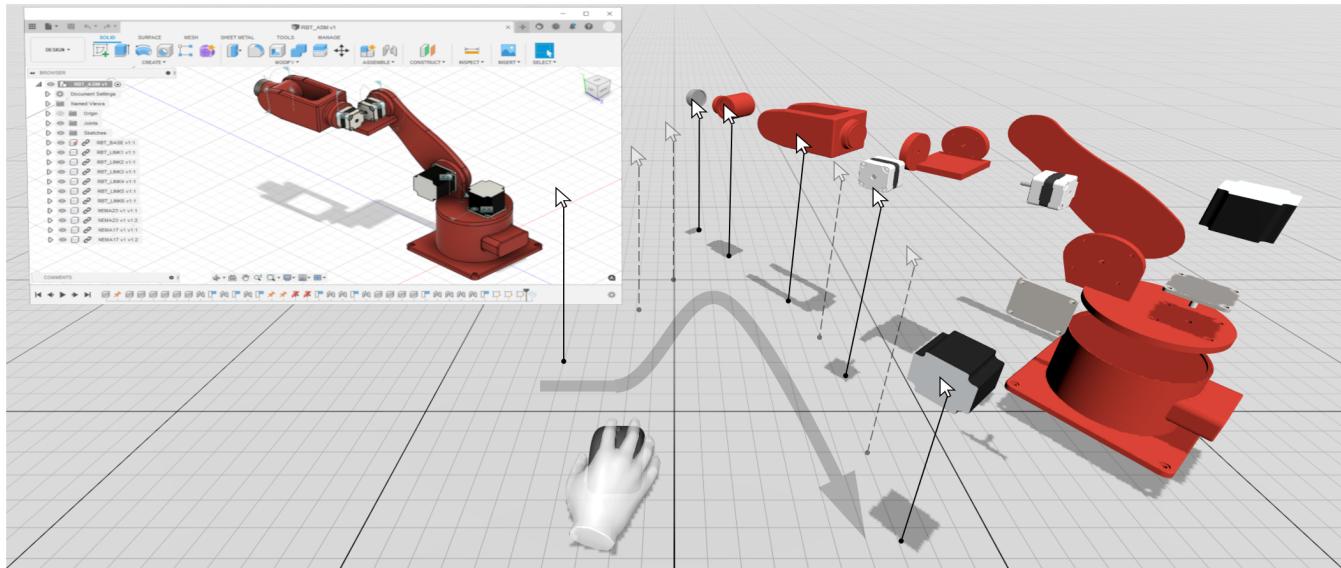


Figure 1: We investigate a technique that integrates a desktop mouse into VR to support productive knowledge work. Our approach uses *Depth-Adaptive Cursor*, a 2D-mouse driven pointing technique for 3D selection with depth-adaptation that continuously interpolates the cursor depth by inferring what users intend to select based on the cursor position, the viewpoint, and the selectable objects. Vertically dropped lines and arrow are added for illustration of depth.

ABSTRACT

Virtual Reality (VR) has potential for productive knowledge work, however, midair pointing with controllers or hand gestures does not offer the precision and comfort of traditional 2D mice. Directly integrating mice into VR is difficult as selecting targets in a 3D space is negatively impacted by binocular rivalry, perspective mismatch, and improperly calibrated control-display (CD) gain. To address these issues, we developed *Depth-Adaptive Cursor*, a 2D-mouse driven pointing technique for 3D selection with depth-adaptation that continuously interpolates the cursor depth by inferring what users intend to select based on the cursor position, the viewpoint, and the selectable objects. *Depth-Adaptive Cursor* uses a novel CD gain tool to compute a usable range of CD gains for general mouse-based pointing in VR. A user study demonstrated

that *Depth-Adaptive Cursor* significantly improved performance compared with an existing mouse-based pointing technique without depth-adaption in terms of time (21.2%), error (48.3%), perceived workload, and user satisfaction.

CCS CONCEPTS

- Human-centered computing → Pointing; Virtual reality.

KEYWORDS

Virtual Reality, 3D pointing, target selection, virtual workspace

ACM Reference Format:

Qian Zhou, George Fitzmaurice, and Fraser Anderson. 2022. In-Depth Mouse: Integrating Desktop Mouse into Virtual Reality. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA*. ACM, New York, NY, USA, 17 pages. <https://doi.org/10.1145/3491102.3501884>

1 INTRODUCTION

Recently there has been an emerging trend of utilizing Augmented Reality (AR) or Virtual Reality (VR) head-mounted displays (HMDs) to enhance office work by extending traditional 2D displays into the third dimension. Effort from both academia and industry has pushed this vision closer to reality, including Varjo Workspace

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 the author(s) 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.

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9157-3/22/04...\$15.00

<https://doi.org/10.1145/3491102.3501884>

[62], Oculus Infinite Office [46], Lenovo ThinkReality Workspace [37], and various VR applications that render the user's computer monitor in VR [16]. In these applications, users can stream their desktop interface into the virtual environment by rendering floating windows or virtual displays that duplicate a desktop environment in VR, creating a virtual workspace that enables users to work productively everywhere with head-mounted displays [26].

The envisioned workspace provides more screen real estate by allowing users to visualize 2D windows and 3D models at the same time. In such display space, users can alternate between 2D and 3D content such as selecting a vertex on a 3D model followed by clicking a button in a 2D menu. Current VR HMDs use handheld controllers with simple ray-casting or midair gestures as their primary input [25]. While these approaches can be intuitive and expressive in free-space, they are not ideal for 2D desktop interfaces due to high fatigue and low precision [8]. Traditional pointing devices (i.e., mice) are accurate and comfortable to use due to their constrained input space and the support of the surface they are used on. Recently, users have been able to connect mice to HMDs such as Oculus Quest or HoloLens [47]. However, the use of mouse is primarily constrained to 2D planes within the 3D space such as browsing websites. Selecting 3D targets that are varied in depth with a regular mouse remains a challenge in VR.

The goal of this work is to extend the current use of a standard, 2D desktop mouse into VR in a consistent manner that leverages users' existing skills, the device's high precision, and low fatigue to facilitate accurate and comfortable 3D target selection in VR. The work addresses three primary problems with general mouse pointing in VR: the *diplopia problem*, the *perspective problem*, and the *sensitivity problem*. Each problem has been previously identified in a display environment different from, but relevant to, VR HMDs.

- The *diplopia problem* was first identified [66] in stereoscopic desktop displays for users experiencing diplopia (i.e. double vision) when aligning the cursor with an object at a different depth. Converging the eyes on the object will produce two images of the cursor (Figure 2 (Top)), causing discomfort with decreased performance [61]. Similarly, VR HMDs provide stereoscopic views and thus are subject to the problem.
- The *perspective problem* was identified within multi-display environments when 2D interaction techniques are applied in 3D without considering the perspective of users [44]. The visibility and overlapping relationship between objects is different based on the user's position and thus the control from the mouse needs to be applied differently based on the viewpoint (Figure 2 (Bottom)).
- The *sensitivity problem* is a common problem in determining the control display (CD) gain for large displays. Low CD gains in large displays cause frequent clutching when travelling to distant objects, while high CD gains make it hard to select small objects due to human precision limitations and device quantization [45]. VR HMDs provide large rendering space. Therefore the mouse's CD gain should not be determined arbitrarily.

To resolve these problems, we developed *Depth-Adaptive Cursor*: a mouse-based 3D pointing technique with depth-adaptation that interpolates the cursor depth from the depths of the objects nearby

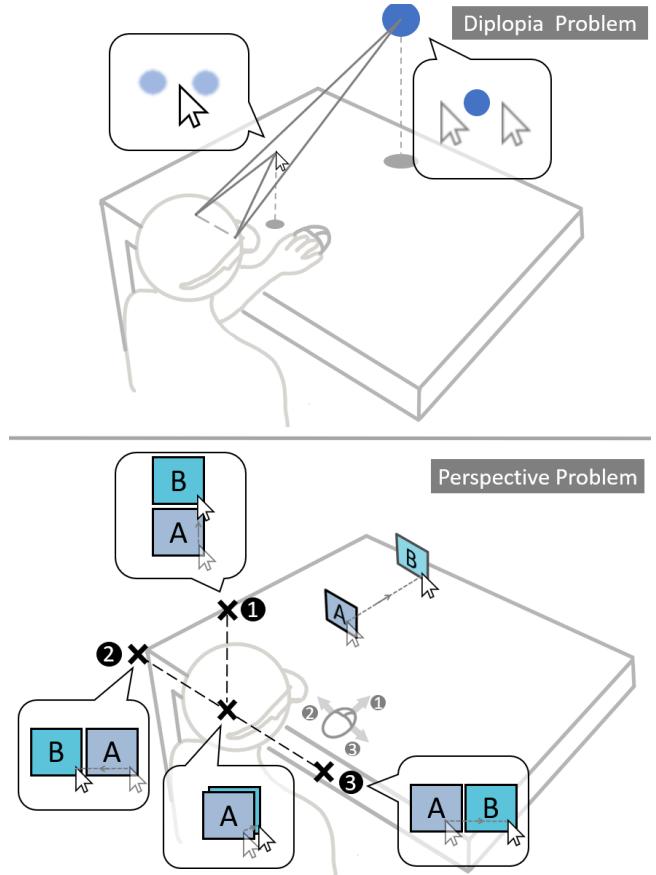


Figure 2: The diplopia and perspective problems for mouse pointing in VR. (Top) A user intends to select a blue sphere and sees double images of the sphere when converging eyes on the cursor, or double images of the cursor when converging on the object, caused by the depth difference between the cursor and the object. (Bottom) A user moves the cursor at object A to select object B. The visibility and overlapping relationship between A and B is different based on the user's position and therefore the control needs to be applied differently. For example, the mouse needs to move upwards at head position 1, left at position 2, and right at position 3.

by inferring what users intend to select based on the cursor position, the viewpoint, and the objects. It integrates the desktop mouse into the virtual workspace with a control mechanism consistent to the desktop mouse. This work has three main contributions: 1) we design a mouse-based 3D pointing technique with depth-adaptation and view-dependency that addresses the *diplopia problem* and *perspective problem*. 2) we adapt a theoretical model from prior work [45] to analytically determine a usable range of CD gains based on the capabilities of the VR HMDs, the mouse, and the expected range of target configurations, addressing the *sensitivity problem*. 3) we present findings from a user study with 16 participants showing *Depth-Adaptive Cursor* outperformed *Perspective Cursor* [44], an

existing mouse-based pointing technique without depth adaptation, in terms of time (20.9%), error (48.3%), perceived workload, and user satisfaction. Our results demonstrated the effectiveness of the depth-adaptation of *Depth-Adaptive Cursor* to enable continuous, comfortable, and accurate target selection for users to work productively in VR.

2 RELATED WORK

Our work is motivated by the emerging trend and the challenges in prior work to support knowledge work in VR, as well as work that explores the use of mice for 3D selection tasks. This work also builds upon existing mouse-based pointing techniques outside of VR that consider the *diplopia*, *perspective*, and *sensitivity problem*.

2.1 Knowledge Work in VR

A number of prior works have explored the use of AR and VR for productive knowledge work. Early work used projectors [54] or stereoscopic displays coupled with head-tracking [11, 60] to instrument and augment the office space. Recent research has extended this concept to HMDs, which can render content anywhere, replacing [26, 49] or augmenting [38, 42] existing physical displays. With VR HMDs, users can stream their desktop content into an immersive environment decoupled from their physical surroundings [48], while with an AR HMD, users can augment the existing office with a large display space [42] or display relevant information at a different depth layer [17]. Enabling expressive, efficient and comfortable input has been an ongoing challenge in the AR and VR workspace. Previous research has concentrated on the spatial gestures with controllers or hands for pointing [63], manipulation [40], and text-entry [27, 56]. While they are effective for free-space interactions, they are not ideal for productive knowledge work, especially for 2D windows with increased fatigue and decreased precision [8, 50]. Other approaches have considered touch input with tablets [9] and wearable devices [58], but these approaches are still limiting to support long hours of productive work. Recent work has considered traditional desktop-based input devices such as mouse and keyboard [42] given that they are optimized for precision and long hours of work [25]. While several approaches have focused on configuring the keyboard for text-entry in VR [27, 56], little work has looked at standard 2D pointing devices such as the mouse, which offers high precision and comfort. They are relatively underexplored but worth consideration for productive knowledge work such as with 3D modelling [11] and visual analytics [64] in VR. This paper aims to fill this gap to provide accurate and comfortable 3D selection in consideration of potential issues with 2D pointing devices such as mouse in the VR workspace.

2.2 Comparison of Input Devices in 3D

Prior research has compared and contrasted the mouse with several input devices for pointing and manipulation tasks. Early work compared mice with 3D input devices such as laser pointers or 3D trackers. The mouse has been found to outperform these devices for common tasks such as 3D pointing task [43] and 3D positioning task [4, 7, 59]. Compared to the limited accuracy of 3D tracking used at the time, the higher resolution of mice could be a plausible

explanation for its superiority in the early studies [59]. Recent advance in 3D tracking with improved accuracy has led to subsequent studies comparing the mouse to other pointing devices such as modern VR controllers and touchscreens. Kovarova et al. has found that the mouse and keyboard outperformed the touch input on a smartphone in translation and rotation tasks [33]. Besançon et al. found that the mouse has advantages of higher accuracy and lower fatigue at the cost of speed compared to tactile and tangible input in 3D positioning tasks [8]. Martel et al. [41] have found players preferred to use the mouse to control in-game functions in VR games. Similarly, Seibert et al. [57] has found the mouse was perceived as more natural and Farmani et al. [19] has found the mouse provided better task performance than VR controllers in 3D shooting games. More recently, Petford et al. [50] compared the mouse pointing with the laser pointing enabled by a motion-capture system and found that the mouse pointing was fastest when the targets do not require the user to move the body, while the laser pointing was superior for targets requiring body movement. Pham et al. found the mouse and pen input were comparable while they both outperformed modern VR controllers [51]. Krichenbauer et al. found that users performed object manipulation faster with mouse in AR over VR [35].

Overall, the mouse has been found to provide better accuracy and comfort compared to other pointing devices, potentially due to user familiarity, high precision, and a supporting surface to reduce fatigue and improve precision [59]. We aim to support productive knowledge work where users can interact with both 2D windows and 3D content within the immersive space. It is important to ensure that they can point and select 2D and 3D targets in a consistent way as they will perform on a regular desktop with low fatigue and high precision. Therefore, based on the advantages of the mouse pointing in previous work, we focus on extending the desktop mouse into the virtual workspace that leverages users' familiarity, the device's high precision, and low fatigue.

2.3 Mouse-based Pointing Techniques

VR HMDs have shared properties with other display environments such as the immersion with large displays, and share many of the same challenges when integrating mouse-based input. These challenges include the *diplopia problem* in stereoscopic displays, the *perspective problem* in multiple-display environments, and the *sensitivity problem* in large displays environment. We look at existing mouse-based pointing techniques in these display environments addressing the challenges in consideration of their shared characteristics with VR HMDs.

2.3.1 Techniques in Stereoscopic Displays. The *diplopia problem* was first identified in stereoscopic desktop displays for users experiencing double-vision when aligning the cursor with an object at a different depth. Early work explored the One-eye Cursor [66] that rendered the cursor only to the dominant eye to resolve the diplopia problem. Further studies found that One-eye Cursor caused greater eye strain and discomfort [55, 66]. To overcome this issue, other techniques such as the Sliding Cursor [60] have been proposed by controlling an invisible cursor on a virtual plane, casting a ray from the viewpoint to the invisible cursor, and displaying a visible cursor at a position where the ray intersects with objects in the scene. The visible cursor slides along the surface of the object and is

therefore always displayed at the object depth, potentially resolving the *diplopia problem* since there is no depth difference between the cursor and object. Binocular Cursor takes a different approach of utilizing the double-vision by combining the images from two eyes to create a converged cursor [36]. Prior work has compared these techniques to the One-eyed Cursor [36, 55, 60] with mixed results depending on the layout of 3D targets [61]. Variations of these techniques have later been applied to Spatial Augmented Reality (SAR) [22, 23], extending the reach of a desktop mouse at the potential cost of performance [22] and the unfavorable effect that changing the head position causes the cursor to move on the object [23].

A major limitation of existing techniques is the absence of ways to determine the cursor depth with depth-varied objects. The *diplopia problem* persists when the cursor travels from one object to another at a different depth with no objects in between. Users have to co-linearly align the cursor, their viewpoint, and the target to get the cursor “snapping” onto the destination object, causing double-vision especially with a small target and a long travel distance. The problem is less salient in the stereoscopic desktop displays with a small and finite display volume. In contrast, VR HMDs have a large display space with stereoscopic capability and thus require techniques to provide depth continuity when the cursor travels in an object-less space.

Conceptually, our work is closest to the most recent work of EZCursorVR [53], which used a mouse for 2D pointing in VR HMDs, and VRMouse [30] which emulated the mouse with VR controllers. However, the prior work primarily investigated 2D pointing with all targets located at the same depth. It is unclear how to determine the cursor depth with depth-varied objects. In addition, techniques such as EZCursorVR [53] attached the cursor to the head, rotating the head causes the cursor to move in the scene, making this approach inconsistent with regular desktop mouse control. We seek to find an approach that can address the *diplopia problem* as well as provide a control mechanism consistent with the desktop mouse.

Techniques in Multi-displays. Early work in multi-display environments introduced several techniques to achieve cross-display cursor transfer, including resolving the discontinuity [5], reducing the long travel distance across displays [6], and using head-tracking to quickly switch the cursor between displays [3]. One of the most influential work is Perspective Cursor [44], which utilizes the user’s perspective to determine the cursor’s position together with the mouse input. When multiple displays are distributed in the space, controlling the mouse to move from one display to another depends on the relative position between the two displays and the user. The visibility and overlapping relationship between A and B depends on the user’s position and thus the mouse needs to move along different directions when the perspective is different (Figure 2 (Bottom)). This *perspective problem* is absent on a single planar display, present in a multi-display environment, and becomes more important in VR HMDs if each virtual display is considered equivalent to a selectable object and as there are multiple selectable objects widespread in VR HMDs. Perspective Cursor [44] and its subsequent refinements [20, 65, 68] addressed the *perspective problem* by adapting to different visibility of the displays. The position and orientation of each display relative to the user’s position determine how the control from the mouse is applied. While it seems

promising to apply Perspective Cursor in VR HMDs by mapping each selectable object in VR as a display unit in the multi-display environment, the approach is designed for 2D interfaces and does not determine the cursor depth when travelling in the display-less (non-planar) space. Applying the Perspective Cursor to VR HMDs requires depth-adaptation that determines the cursor depth based on the objects nearby. Inspired by prior work on perspective-based interactions, we extend Perspective Cursor into the 3D space and interpolate the cursor depth by inferring what users intend to select based on the cursor position, the viewpoint, and the selectable objects in the scene.

Techniques in Large Displays. The *sensitivity problem* is common in determining the CD gain for large displays. Low CD gains in large displays cause frequent clutching when travelling to distant objects, while high CD gains make it hard to select small objects due to hand precision and device quantization [45]. Previous work has concentrated on the trade-off between the precision and sensitivity of mice to avoid repeated clutching when the cursor moves long distances [13, 15, 21]. Approaches to resolve this trade-off have dynamically adjusted the CD gain [21] or the cursor size [15] based on the hand speed to improve precision. Casiez et al. [13] defined a pointing framework to determine a usable range of CD gains based on the capabilities of the mouse, display, and the expected range of target positions and scales. Nancel et al. extended the framework and applied it to other input devices such as a gyromouse and touchpad [45]. They found it difficult to use constant CD gains for various devices for high precision pointing without clutching in large displays. Other work used mice with constant CD gains combined with other input such as body orientation [18], head orientation [53] and eye gaze [39]. In these works, the input from the mouse is usually used as refinement in a coarse-to-fine selection process. We choose not to use similar two-tier approaches to stay consistent with regular mouse control. VR HMDs provide large rendering space similar to the large display environment. Therefore the mouse’s CD gain in VR should not be determined arbitrarily. As there are no available guidelines on how to determine CD gains in VR HMDs, we adapted Nancel’s framework [45] into VR to determine a usable range of mouse CD gains for general mouse-based pointing in VR HMDs.

3 IN-DEPTH MOUSE

The goal of this work is to better integrate a traditional mouse into VR by addressing the *diplopia*, *perspective*, and *sensitivity* problems. To achieve this, we introduce *Depth-Adaptive Cursor* (Figure 3), a mouse-based 3D pointing technique which resolves the *diplopia* and *perspective* problems, and a CD gain tool to compute a usable range of CD gains for general mouse-based pointing in VR to address the *sensitivity* problem. The system pipeline of the proposed approach consists of an offline tool adapted from [45] to analytically determine CD gains, and an online pipeline that updates the cursor position based on the head position, target positions, and the mouse movement (Figure 4). We explicitly seek to find a control mechanism consistent with a regular desktop mouse that leverages users’ existing skills to support the desktop knowledge work. This consistency implies: 1) users should be able to see the cursor clearly with both eyes, 2) the cursor should move continuously in 3D as

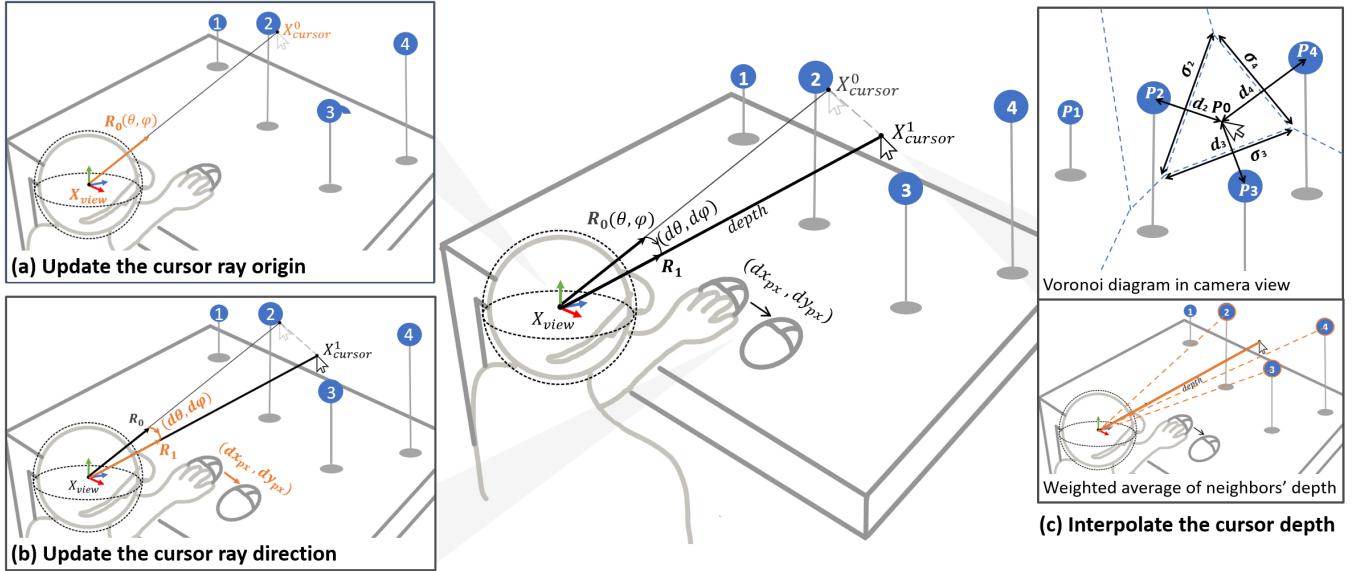


Figure 3: An exemplary diagram of *Depth-Adaptive Cursor* with four objects. The cursor’s position is computed based on a cursor ray R_1 originated from the viewpoint and the depth along the ray. In each frame, we determine the cursor’s new position X_{cursor}^1 by updating R_1 and the depth. (a) We update the origin of a cursor ray based on the current head position X_{view} and previous cursor position X_{cursor}^0 . (b) We update R_1 ’s direction based on the mouse delta motion in angular measures ($d\theta, d\varphi$) converted from its movement (dx_{px}, dy_{px}) in pixels. (c) We determine the cursor’s depth using a Voronoi-based Laplacian interpolation. A Voronoi diagram is generated using the projected points from the cursor ray R_1 (projected as P_0) and four objects ($P_1 \sim P_4$) in the image plane of the viewpoint. The projections of objects $P_2 \sim P_4$ are the natural neighbors of the cursor P_0 . The cursor depth is computed as a weighted average of the natural neighbors’ depths with weights determined by the edge length σ_i and distance d_i to P_0 for each neighbor i .

it does in 2D displays, and 3) the cursor should only be moved by the mouse so that moving the head will not cause any cursor movement. These requirements make a number of existing mouse-based techniques inapplicable, such as One-eyed Cursor [66], Sliding Cursor [60], and EZCursorVR [53]. *Perspective Cursor* [44] is the most promising technique to provide the control consistency in VR naturally resolving the *perspective problem*. *Depth-Adaptive Cursor* is based on *Perspective Cursor* with depth-adaptation to address the *diplopia* and *perspective problem*. Together with the CD gain tool, our approach aims to provide a control mechanism consistent to a regular desktop mouse with seamless transition between the 2D and 3D content in the virtual workspace.

3.1 Depth-Adaptive Cursor

A desktop mouse is a 2D pointing device that moves with 2 degrees of freedom (DoFs) mapped to the horizontal and vertical movement on 2D displays. In VR, the virtual cursor moves with 3 DoFs to select a 3D object. We map the 2 DoFs from the mouse to the horizontal and vertical control in VR, and use a Voronoi-based Laplacian interpolation to determine the depth dimension based on the cursor position, the viewpoint, and the selectable objects. The Laplacian interpolation provides linear continuity [10] in the depth dimension so that the cursor can move continuously in 3D. The control mechanism is different from prior work [23, 53, 60] that combined the head and mouse movement to control the cursor by

attaching it to a virtual plane. We use the head position (the center of two eyes) to determine the geometric relationship of objects such as occlusions so that the user can select objects which are visible from the viewpoint. Based on this relationship, the cursor is solely controlled by the mouse so that moving the head will not cause cursor movement, keeping it consistent with regular mouse control.

3.1.1 Update Cursor Ray. *Depth-Adaptive Cursor* works in the same way as *Perspective Cursor* [44] in the horizontal and vertical dimensions. We use a ray R_0 to represent the cursor’s direction with respect to the viewpoint. At the beginning of each frame, we update the origin of the cursor ray at the current position of the tracked viewpoint X_{view} . The cursor ray R_0 is defined by a line connecting the origin X_{view} to the cursor position X_{cursor}^0 in the previous frame. We convert R_0 to angular coordinates as (θ, φ) in a local spherical coordinate system (dashed-line sphere in Figure 3) that has its origin attached to the head position and its orientation aligned with the world coordinate system. This process is shown in Figure 3(a).

We update the direction of the cursor ray R_0 by detecting the movement from the mouse. Once a delta movement (dx_{px}, dy_{px}) is detected in pixels, it is converted into angular movement $(d\theta, d\varphi)$ in spherical coordinates as:

$$\begin{pmatrix} d\theta \\ d\varphi \end{pmatrix} = \frac{CDgain}{Res_{mouse}} \begin{pmatrix} dx_{px} \\ dy_{px} \end{pmatrix} \quad (1)$$

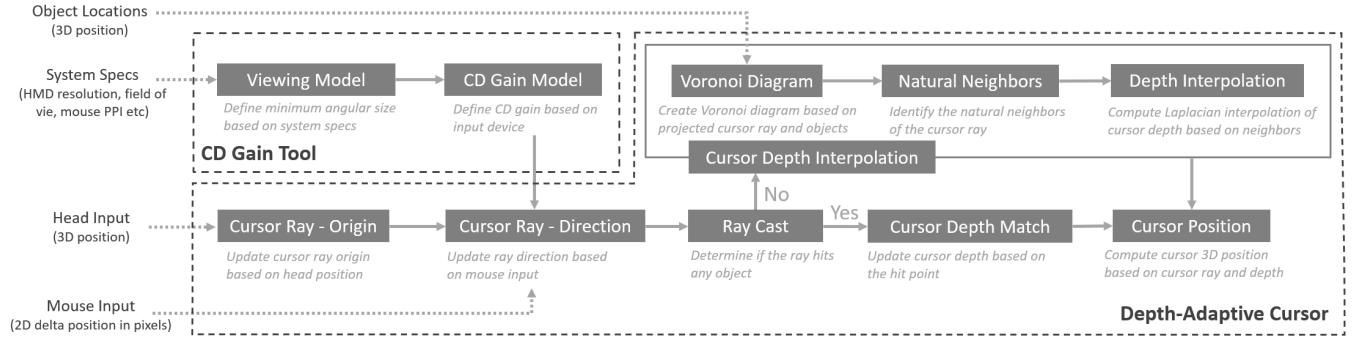


Figure 4: Our system pipeline, including *Depth-Adaptive Cursor*, a mouse-based pointing technique for 3D targets with depth-adaptation and a CD gain tool that analytically determines a usable range of CD gains based on the capabilities of VR HMDs. In each frame, *Depth-Adaptive Cursor* gets updated with a cursor ray and cursor depth. The cursor ray is determined by the current head position and mouse movement. Then we cast the cursor ray in the scene. When the ray hits any selectable objects, the cursor depth is equal to the intersection depth. When the ray does not intersect with any objects in an object-less space, we interpolate the cursor depth based on the depths of its neighbors in a Voronoi diagram.

Res_{mouse} is the mouse resolution in dots per inch. CD_{gain} is the mouse control display gain in degree per inch (DPI). We use a tool to determine CD_{gain} discussed in details in Section 3.2.2. The delta motion ($d\theta, d\varphi$) is added to R_0 , generating a new ray R_1 ($\theta + d\theta, \varphi + d\varphi$) that represents the cursor ray in the current frame (Figure 3(b)).

3.1.2 Ray Cast. We cast the cursor ray R_1 into the scene to find whether it intersects with objects to be selected. If it hits any objects, the cursor *depth* is set to be the depth of the closest intersection. Together with the cursor ray R_1 , the current cursor position X_{cursor}^1 can be computed based on R_1 and the *depth*. If the ray does not intersect any objects, the cursor *depth* will be interpolated based on its natural neighbors of selectable objects in a Voronoi diagram.

$$X_{cursor}^1 = X_{view} + R_1 \cdot depth \quad (2)$$

3.1.3 Cursor Depth Interpolation. This step provides the depth-adaptation when the cursor ray R_1 does not intersect with any objects in the scene. We project R_1 and selectable objects into the camera view. A Voronoi diagram is created based on the projected coordinates in 2D (e.g. $P_0 \sim P_4$ in Figure 3(c) with P_0 as the projected cursor). The *depth* of the cursor is computed based on its N natural neighbors in the Voronoi diagram (e.g. $P_2 \sim P_4$ in Figure 3(c)). To compute the cursor *depth*, we apply the Laplacian interpolation [10] as a weighted average of the neighbors' depths, defined as the Euclidean distance $d(P_i, X_{view})$ between the 3D position of a natural neighbor i and viewpoint X_{view} , for all N natural neighbors of P_0 in the Voronoi diagram:

$$depth = \sum_{i=1}^N \lambda_i \cdot d(P_i, X_{view}) \quad (3)$$

When computing the weighted average in equation 3, each weight λ_i is determined based on the proximity between a neighbor P_i and the cursor P_0 . The proximity metric $\hat{\lambda}_i$ is a ratio between the length of the shared edge σ_i and the Euclidean distance d_i between P_0 and its neighbor P_i in equation 4. We normalize $\hat{\lambda}_i$ for all N

neighbors to obtain the weight λ_i in the Laplacian interpolation. The Laplacian interpolation results in linear continuity in *depth* when the cursor travels between objects. We chose the Laplacian interpolation as it has been found to be more efficient compared to others such as the Sibsonian approach for natural neighbor interpolation [32]. An example is shown in Figure 3 (c).

$$\lambda_i = \frac{\hat{\lambda}_i}{\sum_{j=1}^N \hat{\lambda}_j}, \quad \hat{\lambda}_i = \frac{\sigma_i}{d(P_i, P_0)} \quad (4)$$

3.2 CD Gain Tool

We use a theoretical model adapted from prior work [45] to analytically determine a usable range of CD gains based on the capabilities of the mouse, HMDs, and object configurations. We first determine the object configuration using a viewing model considering the limitation of the human visual system and HMDs.

3.2.1 Viewing Model. The human visual system has limited acuity to discern small details with precision. The minimum visual angle for a visual stimulus to be distinguishable with human eyes in normal vision is 1/60 deg, and 1/12 deg to be legible. In the virtual world, the resolution of HMD places additional constraints to display the smallest object (1 pixel), and a legible character (8 pixels), which can be converted into the angular size of 0.05 deg and 0.37 deg respectively in an HMD with the resolution of 20.58 pixels per degree (PPD) such as an Oculus Quest 2 [67]. As we aim to support desktop interfaces in HMDs, we choose the largest constraint of 0.37 deg as the lower bound to ensure the objects are legible in the virtual workspace. Note that this constraint can be smaller with an HMD that has a higher resolution. For example, a Hololens 2 with 46.5 PPD will have a lower bound of 0.17 deg. The constraint of the angular size does not directly provide a usable range of positions for us to render the objects in the scene. The object configuration can be determined by a viewing model from [45] based on the viewing distance D , the angular distance α , the target size W , and its angular size β as shown in Figure 5 (Left):

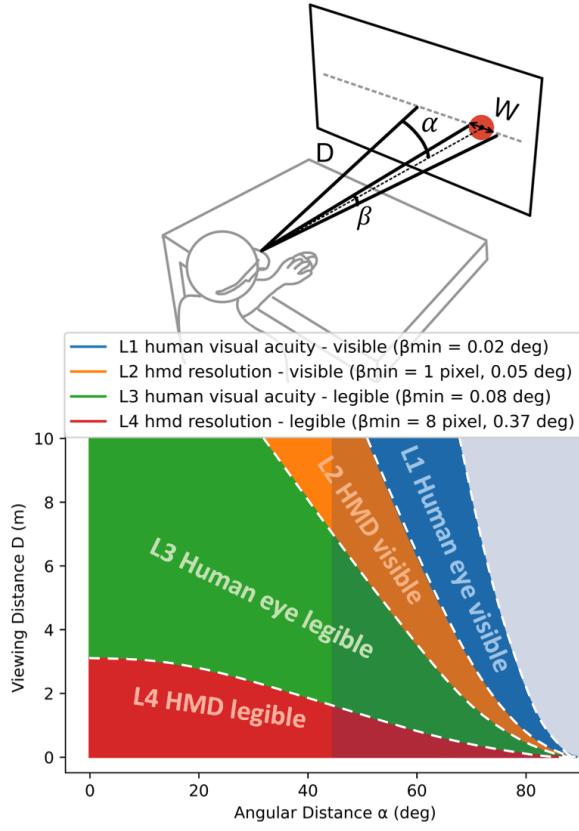


Figure 5: (Top) A viewing model adapted from prior work [45] to determine the object configuration in VR based on the viewing distance D , the angular distance α , the target size W , and its angular size β . **(Bottom)** An example of usable ranges of object locations for a 2 cm object to be perceivable (L1-2) or legible (L3-4) in HMDs. L1 and L3 are based on human vision capability while L2 and L4 are based on the specs of Oculus Quest 2 [67]. Shaded regions are outside the field of view of the HMD. We use the L4 region to determine locations for targets in the study.

$$\tan^{-1}(\tan(\alpha) + \frac{W}{2D}) - \tan^{-1}(\tan(\alpha) - \frac{W}{2D}) = \beta \quad (5)$$

This equation helps us to determine where we can safely render an object with the size of W if we want the object to be perceivable in an HMD. Using this equation, we show an example of simulated results for a 2 cm object. In Figure 5 (Right), each colored region represents a usable range of the angular distance α and viewing distance D for a 2 cm (W) target to be perceivable (L1-2) or legible (L3-4) in an HMD with the specs of an Oculus Quest 2 [67]. Shaded regions are outside the field of view. We use this viewing model to design the objects in our user study and ensure objects within the field of view are inside the usable range to be legible (i.e., red region for 2 cm objects) in the experiment.

3.2.2 CD Gain Model. Prior work [13, 45] provides an approach to compute a usable range of CD gains for input devices on 2D displays.

To adapt it for 3D displays, we convert the parameters into angular coordinates so that the CD Gain of the mouse is described in *degree per inch* (DPI) instead of the standard *dots per inch* (dpi). Similarly, we describe the HMD display resolution in *pixel per degree* (PPD), the maximum travel distance A_{max} in *degree*, and the minimum object size W_{min} in *degree*.

The lower bound CD_{min} is determined as the ratio between the maximal angular travel distance A_{max} of a cursor to select a target in the VR space, and the maximal input operating range OR in the physical space. When the CD gain of a device is lower than CD_{min} , users need to clutch multiple times to move to a target:

$$CD_{min} = \frac{A_{max}}{OR} \quad (6)$$

The upper bound CD_{max} is determined by both device precision CD_{qmax} and human hand precision CD_{lmax} . CD_{qmax} is the upper bound based on the minimal input movement that can be sensed to move the cursor by one pixel. It is defined as the ratio between the minimal display unit Res_{hmd} , determined by the display resolution, and the minimal input movement that can be sensed (Res_{device}). CD_{lmax} is the upper bound based on the minimal input movement that can be performed by human hands. It is the ratio between the smallest target size W_{min} and the minimal human hand input Res_{hand} . When the CD gain of a device is greater than either CD_{qmax} or CD_{lmax} , some pixels become unreachable caused by insufficient device precision or human precision.

$$CD_{max} = \min\left(\underbrace{\frac{Res_{device}}{Res_{hmd}}}_{CD_{qmax}}, \underbrace{\frac{W_{min}}{Res_{hand}}}_{CD_{lmax}}\right) \quad (7)$$

Note that CD_{min} and CD_{max} are determined independently, indicating the CD_{min} can be greater than CD_{max} . In this case, users will encounter at least one of the clutching and precision problems that should be avoided. Prior work [45] has found several input devices are impractical to use with CD_{min} greater than CD_{max} and required pointer acceleration as a solution in large displays.

We use these equations to compute the thresholds of CD gains for a mouse and VR controllers in three HMDs commonly used in the literature, including HTC Vive Pro, Oculus Quest 2, and Hololens 2 (Table 1). Calculations are based on a standard mouse with 1000 dots per inch (dpi), a maximal travel distance A_{max} of 180 deg, and a minimum target size W_{min} constrained by the resolutions of HMDs to be legible with the minimum size of 8 pixels.

For the mouse in all three HMDs, we found CD_{min} smaller than CD_{max} . A constant CD gain can be determined within an interval of [15.24, 76.39] in Vive Pro, [15.24, 48.59] in Quest 2, and [15.24, 21.50] in Hololens 2. Note that the interval is wider in an HMD with a lower resolution as Vive Pro has the resolution of 13.09 PPD, Quest 2 has 20.58 PPD, and Hololens 2 has 46.51 PPD [67]. These intervals are based on an operating range of 30 cm. When the mouse is operated in a much smaller space such as 10 cm, CD_{min} increases to 45.72. In this case, the range of CD gains shrinks as [45.72, 76.39] in Vive Pro, and [45.72, 48.59] in Quest 2. If users in Hololens 2 have a limited physical space of 10 cm to operate the mouse, CD_{min} will be greater than CD_{max} and the pointer acceleration should be considered in high resolution HMDs similar

Device	Res_{device}	Res_{hand}	OR	CD_{min}	$CD_{max} (q_{max}, l_{max})$		
				HTC Vive Pro	Oculus Quest 2	HoloLens 2	
Mouse	1000 dpi	0.2 mm [1, 13]	30 cm [13]	15.24	76.39 (76.39, 77.62)	48.59 (48.59, 49.37)	21.50 (21.50, 21.84)
VR controller	0.17 deg [2]	0.53 deg [45]	180 deg	1.0	0.45 (0.45, 1.15)	0.29 (0.29, 0.73)	0.13 (0.13, 0.32)

Table 1: CD gain thresholds for a mouse and VR controllers for three common HMDs. Results are based on a standard mouse with the resolution of 1000 dots per inch (dpi), a maximal travel distance A_{max} of 180 deg, and a minimum target size W_{min} constrained by the resolutions of HMDs to be legible with the minimum size of 8 pixels. For a mouse in all three HMDs, CD_{min} is smaller than CD_{max} . A constant CD gain can be chosen between 15.24 and 76.39 in Vive Pro, and between 15.24 and 48.59 in Quest 2, and between 15.24 and 21.50 in HoloLens 2. CD_{min} of VR controllers is greater than CD_{max} , indicating the insufficient precision for VR controllers to select a minimum size of 8 pixels in the HMDs. We use the results of the mouse in Quest 2 to determine the CD gain for the mouse-based pointing techniques evaluated in our experiment.

to HoloLens 2. Therefore, determining the mouse’s CD gain depends on the operating range of the users’ physical environment.

As a reference, we also computed the thresholds of CD gains for VR controllers. Similar to prior work in large display environment [45], we found CD_{min} greater than CD_{max} , indicating the insufficient precision for VR controllers to select a minimum target size of 8 pixels in all three HMDs. Although HoloLens 2 uses midair gestures rather than VR controllers for selection, we assume the VR controllers’ tracking precision of 0.17 deg [2] would be an optimistic estimate for the hand-tracking in HoloLens 2. In all three HMDs, CD_{qmax} is the bottleneck causing CD_{max} lower than CD_{min} , indicating the insufficient tracking precision of VR controllers. While it is possible to use high fidelity tracking such as motion capture systems with a higher Res_{device} to make CD_{qmax} larger than CD_{min} , the CD_{lmax} caused by the human hand tremors will still be a problem with a value of 0.73 (Quest 2) and 0.32 (HoloLens 2) lower than CD_{min} , showing insufficient precision of human hands to select a minimum size of 8 pixels with VR controllers.

4 EXPERIMENTAL EVALUATION

We conducted a user study comparing the *Depth-Adaptive Cursor* to the *Perspective Cursor* [44] to validate the effectiveness of *Depth-Adaptive Cursor* in two tasks, including a 3D pointing task and a data analysis task. We used the 3D pointing task to evaluate the efficiency and accuracy of *Depth-Adaptive Cursor* in a controlled environment, and used the data analysis task to evaluate the utility in a more natural and ecologically valid context. Together the two tasks help us to investigate the effectiveness and understand the perceived characteristics of *Depth-Adaptive Cursor*.

4.1 Pointing Techniques

The primary independent variable used in both tasks is the pointing *Technique* with two levels: *Depth-Adaptive Cursor* (DAC) and *Perspective Cursor* (PC). Both DAC and PC are applied to a regular desktop mouse. In both cases, we dynamically resize the cursors based on the depth to ensure that they have a constant angular size. Both DAC and PC move continuously in the same way in the horizontal and vertical dimensions, controlled by the mouse’s 2 DoF movement. The only difference between DAC and PC is the depth-adaptation. *Depth-Adaptive Cursor* automatically adapts

to the depths of the objects nearby and moves continuously in the depth dimension (Figure 13 blue line). *Perspective Cursor* does not adapt to the depths of the objects nearby and therefore snaps onto an object depthwise upon entering it. When travelling in an object-less space, it keeps the depth of the previously selected object (Figure 13 orange line).

4.2 Participants and Procedure

We recruited 16 remote participants (all males) between 19 and 51 years old (averaged 38.3 years) from within Autodesk. Each compensated with a \$25 gift card. All participants are knowledge workers with typical computer-based roles. All but one were right-handed. All participants had VR experience, with a mean self-reported expertise of 2.9 on a 1-4 scale, and all participants frequently used mice. Participants used their own VR devices (Oculus Quest 2) and a Bluetooth mouse to run the study application, which was developed in Unity. Two participants used Logitech MX master and one used Microsoft Precision Mouse. The remaining 13 participants were provided with a Logitech M720 mouse to use in the study. All mice were set to the resolution of 1000 dpi in the study. The study used a within-subject design so that each participant performed two tasks using DAC and PC. The order of *Technique* was counter-balanced.

The study was conducted remotely in participants’ home offices and supervised by an experimenter via a video conference call. Participants filled out a consent form after verbal explanations of the study. They were guided to set up the experimental environment by removing obstacles on their desk and take a seated position that they felt comfortable in and had enough space (30 cm x 30 cm) to operate the mouse, which gave them ± 15 cm to move the mouse horizontally and vertically. Considering their hands might deviate from the center during the study, we took a conservative estimate of 12 cm as the operating range, resulting in a CD_{min} of 38 and a CD_{max} of 48.59 based on Equation 6 and 7. Within this range, we choose a constant CD gain of 38 DPI applied to both DAC and PC to ensure users can select a minimum size of 8 pixels (0.37 deg in Quest 2) without the need of clutching in an operating range of 12 cm. Then they started to conduct the 3D pointing task. At the end of the 3D pointing task per *Technique*, they filled in a usability questionnaire and a NASA TLX questionnaire [28]. Then they proceeded to the data analysis task and conducted the

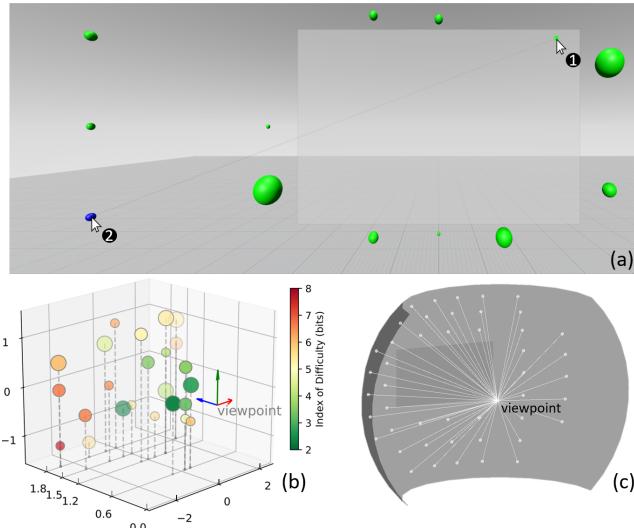


Figure 6: Setup of the 3D Pointing Task: (a) The task is to select an origin 2D target (a square on a 2D plane) and then again in a destination 3D target (a blue sphere in a 3D space). We use the combination of 2D and 3D targets to represent the hybrid layout of a virtual workspace with both 2D and 3D content. (b) shows the placement of the 3D targets varied in Index of Difficulties with three sizes: 0.02 m, 0.04 m, and 0.06 m. Targets vary in distances and are evenly distributed at the depth of 0.6 m, 1.2 m, 1.5 m, and 1.8 m. Their directions are randomly generated from a spherical grid (c).

task per *Technique*. Once they completed the data analysis task, they completed a demographic questionnaire followed by a short semi-structured interview. They were asked if they were able to notice any difference between the techniques, as well as about their preference between techniques. It took approximately 15 minutes to complete the 3D pointing task and data analysis task respectively, with approximately 40 minutes in total for the study.

4.3 Data Analysis

We conducted a repeated-measures ANOVA with significance values reported in brackets for $p < .05(*)$, $p < .01(**)$, and $p < .001(***)$ respectively. Effect sizes are reported as partial eta squared (η_p^2). Numbers in brackets indicate mean (M), median (Med), and standard error (SE) for each respective measurement. When the assumption of sphericity was violated, we applied Greenhouse-Geisser correction. The post-hoc analysis was conducted using pairwise t-tests with Bonferroni corrections. When the assumption of normality is violated, we used a non-parametric alternative of the Wilcoxon signed-rank test to compare the two levels of *Technique*.

4.4 Experiment 1: 3D Pointing Task

4.4.1 Design. To simulate the pointing in a virtual workspace which has 2D windows and 3D objects (Figure 1), we designed an experimental task that required users to toggle between a 2D plane and 3D objects. Participants were instructed to click on an

origin 2D target (a blue square on a 2D plane) and then again in a destination 3D target (a blue sphere in the 3D space) as fast as possible without sacrificing accuracy (Figure 6 (a)). Participants first saw the 2D target highlight in blue. After they clicked on it, the 3D target would highlight in blue with a semi-transparent line connecting to the origin 2D target. They followed the line to select the destination 3D target. The line is provided to minimize the searching time as it could be difficult to find and select a small target in VR's large display space. When a 3D target was not selected after 10 seconds, the trial would time out with a visual notification and the next trial of the origin 2D target would appear.

Participants were given 5 practice trials per *Technique* with the same configuration as the formal trials. They practiced until they felt comfortable to proceed. In the formal session, there were 24 pairs of 2D/3D targets. 2D targets were 0.015 m squares (representing buttons or icons in the desktop applications) randomly placed on a plane (1 m x 0.6 m) that is 0.9 m away from the view. The directions of 3D targets were randomly generated on a spherical grid (Figure 6 (c)) with 20-degree horizontal and 10-degree vertical intervals. Their depths were evenly distributed at 0.6 m, 1.2 m, 1.5 m, and 1.8 m. There are always 8 small (0.02 m), 8 medium (0.04 m), and 8 large (0.06 m) targets. Each target represents a unique Index of Difficulty (ID) with a different target distance ranging from 0.36 m to 3.12 m generated randomly based on the position of 2D and 3D target and was repeated three times (Figure 6 (b)). To avoid memorization from repetition, we divided the three repetitions into three blocks and randomized the sequence of targets in each block. Participants took a break between blocks to mitigate fatigue. Each participant completed a total of 72 (24 x 3) formal trials per *Technique* that yielded a total of 144 trials. For each (non-timeout) trial, we recorded the completion Time to click inside the 3D target. We also recorded the hit/miss information to compute the Error Rate as the ratio between the number of clicks outside targets and the total number of clicks. We collected subjective feedback using a usability questionnaire and a NASA TLX questionnaire [28] per *Technique*.

4.4.2 Results. The data on *Time* and *Error Rate* included only non-timeout trials (2266 of 2304 total trials, or 98.4%). Questionnaire data was analyzed from all participants.

Time. Data on completion time did not meet the assumption of sphericity. A repeated measures three-way ANOVA ($2 \text{ Technique} \times 3 \text{ Target Size} \times 4 \text{ Target Depth}$) with Greenhouse-Geisser correction was performed. We found main effects (Figure 7) for completion time of *Technique* ($F(1, 15) = 57.52$, $p < .001$, $\eta_p^2 = 0.793$), *Target Size* ($F(1.46, 21.9) = 131.32$, $p < .001$, $\eta_p^2 = 0.897$), and *Target Depth* ($F(1.69, 25.39) = 86.12$, $p < .001$, $\eta_p^2 = 0.852$). The mean completion time for DAC ($M = 1.872\text{s}$, $SE = 0.048\text{s}$) was 21.21% lower (*** than PC ($M = 2.376\text{s}$, $SE = 0.060\text{s}$). We found two-way interaction effects of *Technique* \times *Target Size* ($F(1.64, 24.61) = 19.4$, $p < .001$, $\eta_p^2 = 0.564$), *Technique* \times *Target Depth* ($F(1.72, 25.86) = 24.52$, $p < .001$, $\eta_p^2 = 0.620$), and *Technique* \times *Target Size* ($F(3.27, 49.02) = 17.47$, $p < .001$, $\eta_p^2 = 0.538$). We found a three-way interaction effect of *Technique* \times *Target Size* \times *Target Depth* ($F(3.21, 48.14) = 6.13$, $p = .001$, $\eta_p^2 = 0.290$), however given the relatively small effect size this interaction is not further analyzed.

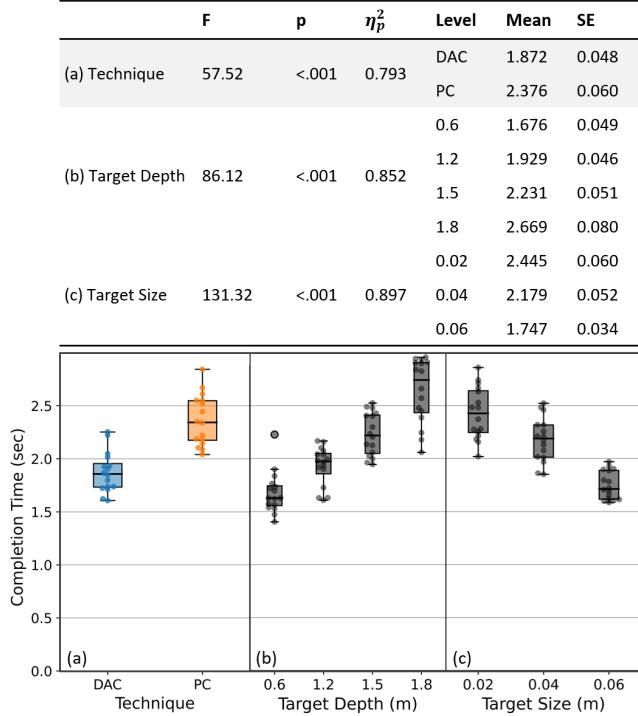


Figure 7: The main effects for 3D targets completion time in the 3D Pointing Task. Boxplots show the main effect ($p < .001$) on (a) Technique , (b) Target Depth , and (c) Target Size . Each dot represents the mean completion time for each participant. The mean completion time for DAC was 21.21% lower than PC.

A post-hoc analysis of the main effect *Target Depth* and *Target Size* using paired t-test with Bonferroni corrections shows significant differences (***) between all four depths (Figure 7 (b)), as well as between all three sizes (Figure 7 (c)). A post-hoc analysis of the two-way interaction effect *Technique x Target Depth* (Figure 8 (a)) shows significant differences between DAC and PC when *Target Depth* is 1.2 m ($t = -3.555$, $p < 0.05$), 1.5 m ($t = -8.471$, $p < 0.001$), and 1.8 m ($t = -6.077$, $p < 0.01$). It shows significant differences (**, ***) with PC between all four depths, except for the comparison between 1.5 m and 1.8 m. It also shows significant differences (***) with DAC between 1.8 m and all three other depths. A post-hoc analysis of the two-way interaction effect *Technique x Target Size* (Figure 8 (b)) shows significant differences between DAC and PC when *Target Size* is 0.02 m ($t = -5.155$, $p = 0.001$), 0.04 m ($t = -7.314$, $p < 0.001$), and 0.06 m ($t = -6.326$, $p < 0.001$). It also shows significant differences (**, ***) between all three sizes with DAC and PC respectively. These interaction effects indicate that DAC and PC were affected differently by *Target Size* and *Target Depth* , with DAC enabling faster selections for targets that were farther away and smaller (Figure 8).

Error Rate. Data on *Error Rate* did not meet the assumption of normality. Wilcoxon signed-rank test was performed. There was a significant difference between levels of *Technique* ($W = 119$, $p <$

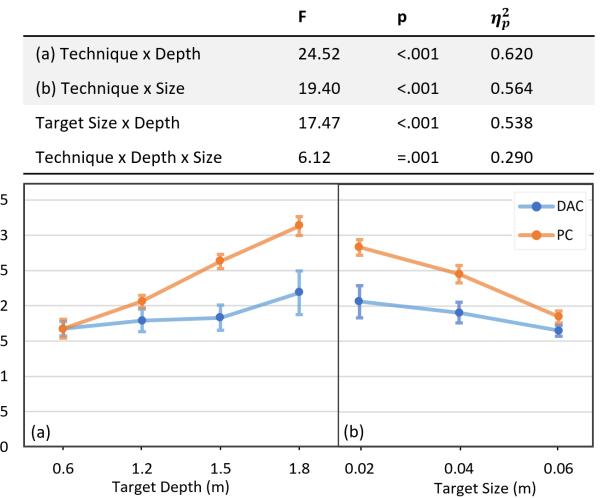


Figure 8: The interaction effects for 3D targets completion time in the 3D Pointing Task. Line graphs show the two-way interaction effects ($p < .001$) of (a) Technique x Target Depth and (b) Technique x Target Size . Error bars represent 95% confidence intervals. The interaction effects show that DAC is less influenced by Target Depth and Target Size compared to PC and enables faster selections for small and distant targets.

.01). The median *Error Rate* for DAC ($Med = 7.09\%$, $SE = 1.14\%$) was 48.25% lower (**) than PC ($Med = 13.7\%$, $SE = 5.36\%$).

Questionnaire. A Wilcoxon signed-rank test was performed, and the results from the usability questionnaire (Figure 10 (a)) showed a significant difference between techniques for efficiency ($W = 66.0$, $p < .01$), consistency ($W = 52.5$, $p < .05$), comfort ($W = 55.0$, $p < .01$), and easiness on small objects ($W = 131.5$, $p < .001$) and distant objects ($W = 134.0$, $p < .001$), but not for sensitivity ($W = 27.0$, $p = .608$). Results from the NASA TLX questionnaire (Figure 10 (b)) showed significant difference for mental demand ($W = 90.5$, $p < .05$), physical demand ($W = 81.0$, $p < .05$), effort ($W = 115.0$, $p < .01$), performance ($W = 105.0$, $p = .001$), and frustration ($W = 96.0$, $p < .01$), but not for temporal demand ($W = 62.5$, $p = .548$).

4.5 Experiment 2: Data Analysis Task

4.5.1 Design. We used a representative data analysis task to evaluate the utility of DAC in a natural and ecologically valid context as compared to the controlled pointing study. The task is adapted from prior work [34] that required users to analyze 3D scatter points data in a virtual workspace. The task was to choose three soccer players from 50 players in the EA Sports FIFA players dataset (sofifa.com), including a striker superior in attacking, a defender superior in defending, and a midfielder balanced between attacking and defending. A limited budget is provided together with the wages, the attacking scores, and the defending scores of players. We use a sphere to represent a player plotted in three dimensions of the wage, attacking and defending score (Figure 9 (b)). Similar to [34], we provided a 2D bar chart of player types. Users clicked

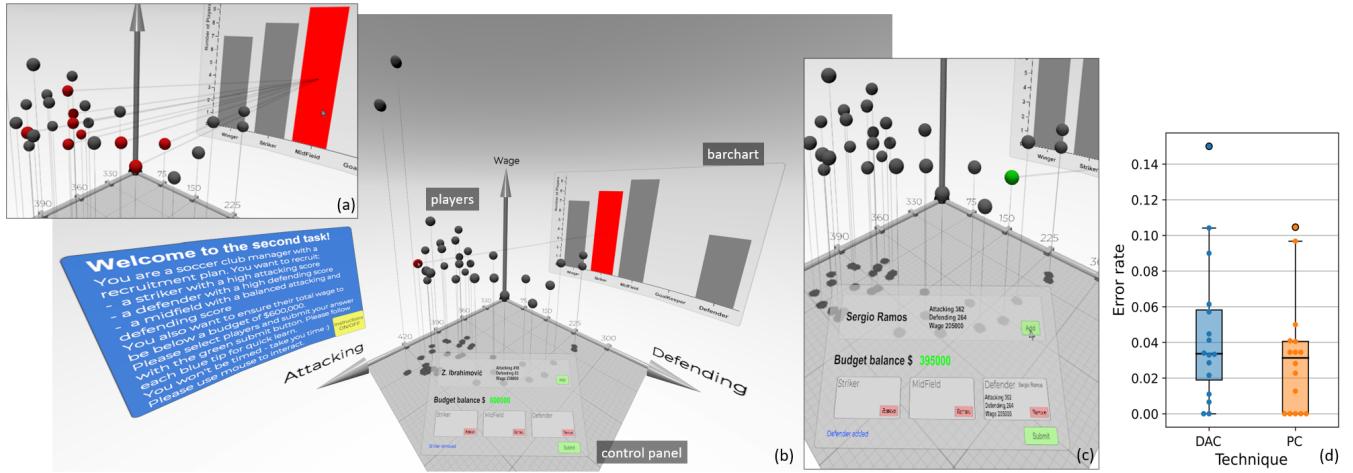


Figure 9: Setup of the Data Analysis Task: the task is to select three soccer players (a striker, a midfielder, and a defender) with a limited budget and corresponding requirements on players' attacking and defending scores. (b) Each player is represented as a sphere scattered in three dimensions of the wage, attacking and defending score. Selecting a sphere shows the player's specs in a control panel and a line linked to the player's role in a bar chart. (a) Users can also select a bar to highlight all players in the same role linked to the bar. (c) Users can add or remove their answers before submitting them. (d) Results of error rate with DAC and PC.

on the sphere to highlight and link to the bar (Figure 9 (b)). They could also click on the bar chart to highlight a type of player in the scatterplot linked to the bar (Figure 9 (a)). They previewed and added the currently selected player in a 2D control panel with a budget balance calculated based on their choices (Figure 9 (c)). They could also remove existing choices and submit the answer in the control panel. They completed the task when their choice of the striker, the defender, and the midfielder with the attacking score, the defending score, and both the attacking and defending score were above required thresholds. The values of thresholds were hidden from the users. We designed the task to encourage frequent toggling between the 2D and 3D objects by clicking on the spheres, the bar chart and the control panel. Before the task per *Technique*, participants were provided with an instruction page explaining the task and were told that they would not be timed (Figure 9 (b)). To prevent them from memorizing the layout and answers that would trivialize the task, we used the 1-50 top players dataset (Figure 9 (b)) the first technique and the 50-100 top players dataset for the second technique with a limited budget depending on the order of techniques. Each player is represented as a 0.08 m sphere placed in a coordinate system shown in Figure 9 (b). Participants could submit their answers multiple times. For each *Technique* condition, we recorded the hit and miss information to compute the ratio (*Error Rate*) between the number of clicks outside any clickable objects (bars, buttons, and spheres) and the total number of clicks.

4.5.2 Results. Data on *Error Rate* did not meet the assumption of normality. Wilcoxon signed-rank test was performed. Results are shown in Figure 9 (d). We did not find a significant difference ($W = 76.0$, $p = .379$) between DAC ($Med = 3.37\%$, $SE = 1.02\%$) and PC ($Med = 3.13\%$, $SE = 0.81\%$).

Overall, 15 of 16 participants preferred DAC over PC based on their experience in both tasks in the post-study interview. They reported DAC was easy (9), natural (3), comfortable (2), predictable (1), and enjoyable (1) to use. In the 3D pointing task, all participants agreed that there was a noticeable difference between DAC and PC. 11 of them mentioned they performed better in DAC. In the data analysis task, most participants (13 of 16) did not notice any difference between DAC and PC. 6 of them explained they concentrated on analyzing the data and did not even notice that the techniques have changed in the two conditions.

5 DISCUSSION

Through the evaluation of *Depth-Adaptive Cursor* across two tasks, we have found that it can achieve a significant reduction in error, workload, and time spent in 3D target selection. While we observed the performance gains in the 3D pointing task, we did not find improvement in the data analysis task.

5.1 Small and Distant Objects

Our results show that it is more difficult to select small and distant objects without depth-adaptation in the 3D pointing task. In the questionnaire, participants rated it significantly more difficult to select small and far objects in PC than DAC (Figure 10 (a)). In the interview, 9 of 16 participants provided similar comments, for instance: “(P5) it felt much harder to select things, especially when they were far away and small”, and “(P16) it was a lot harder to get far away objects”. To select difficult targets, they took different strategies by closing one eye (3), moving the head (3), or wiggling the mouse (2) until the cursor aligns with the target. Three of them further mentioned experiencing double vision when using PC, indicating the *diplopia problem*. Without depth-adaptation, users

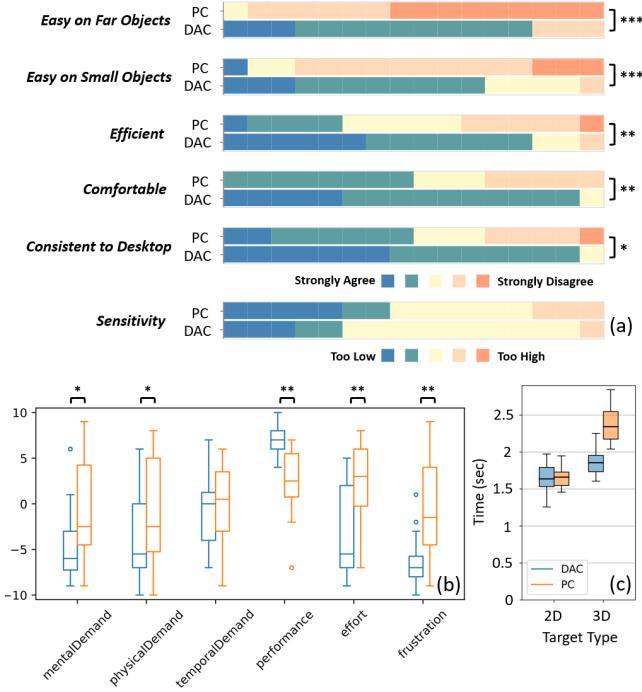


Figure 10: (a) Participants' rates on the usability of DAC and PC from -2 “Strongly Disagree” to 2 “Strongly Agree”. They reported answers on the easiness to select far objects (***) , the easiness to select small objects (***) , the efficiency (**), the comfort (**), the consistency of the control compared to a desktop mouse (*), and the perceived sensitivity of the control. (b) Participants' answers on the perceived workload in the NASA TLX questionnaire. (c) The completion Time for 2D and 3D targets in the 3D pointing task. The performance on 3D targets in DAC is more consistent to the 2D targets group compared to PC.

need to carefully align the cursor, target, and viewpoint on the same line to select the target. The *diplopia problem* makes it difficult to select small and distant targets with PC.

To understand the difficulty of selecting small and distant targets, we used Fitts' law as a predictive model on the Index of Difficulty (ID). We used the Shannon formulation of Fitts' law ($MT = a + b \cdot \log_2(A/W + 1)$) with angular measures (angular sizes and distances) as they have been found to perform better than their linear counterparts [31, 50]. We found that the difference in completion time between DAC and PC is larger on objects with high IDs (Figure 11). The correlations (R^2) of regression for DAC and PC are 0.82 and 0.78, which are relatively lower than 0.9 in other studies [60] using standard methods. One possible explanation is the visual search time. While we provided a line pointing to the 3D target to minimize the search time, it still required participants to rotate the head to find the target, introducing additional time to determine what to select. Another potential factor is the diplopia problem in PC. As participants needed to carefully align the cursor,

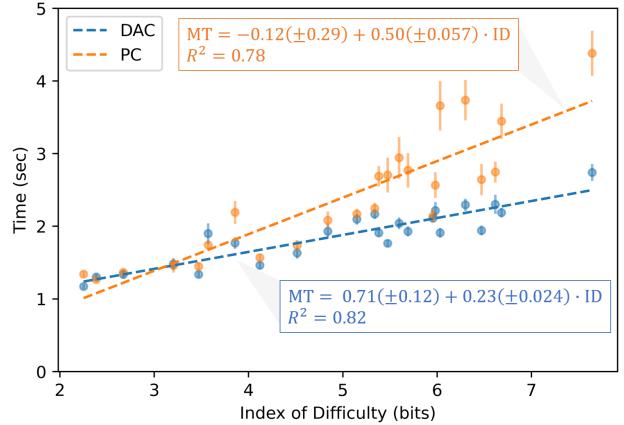


Figure 11: Fitts' law regression lines for DAC and PC in the 3D pointing task. The difference in Time between DAC and PC is larger on objects with high IDs. Error bars represent the standard error for all participants.

target, and viewpoint on the same line for the selection, it would require extra time to wiggle the mouse until the cursor snapped onto the object, which could potentially explain the lower correction in PC (0.78) than in DAC (0.82), particularly on targets with high IDs that appear to deviate more from the regression line compared to others with low IDs in PC (orange line in Figure 11).

Similar to the completion time, the difference in accuracy is more evident on small targets with large depths. To understand its effect on accuracy, we normalized all targets and their associated clicks collected in the 3D pointing task into angular measures for DAC and PC respectively. Selections (red dots in Figure 12) inside the normalized target (green circle in Figure 12) are successful selections while selections outside the target are misses. Comparing Figure 12 (Left) and (Right), the effect of target size and depth appears to be more evident in PC than DAC with more erroneous selections for small and distant targets.

The superiority of *Depth-Adaptive Cursor* on high ID targets could potentially explain the task-related performance. Our quantitative results show that DAC significantly reduced error in the 3D pointing task, but not in the data analysis task. Similarly in the interview, while all participants reported noticeable differences between DAC and PC in the 3D pointing task, 13 of 16 participants did not even notice any difference in the data analysis task. The divergent results could be caused by different target configurations. We collected 1385 selections in the data analysis task with 37% of them performed on 2D selectables (buttons and bars in 2D windows). Since the two techniques work in the same way on 2D targets, we assume the performance difference is less observable in the data analysis task compared to the 3D pointing task. Additionally, the 3D targets in the data analysis task are also easier to select with lower target IDs. In the 3D pointing task, we used small 3D targets (0.02 m, 0.04 m, and 0.06 m) scattered in space, resulting in a range of IDs from 2 to 8 bits (Figure 11). In the data analysis task, we used larger 3D targets (0.08 m) confined in a limited space. The mean ID of all selections collected in the data analysis task is 2.66 bits ranging from 0.27 to 5.25 bits. This indicates that depth-adaptation

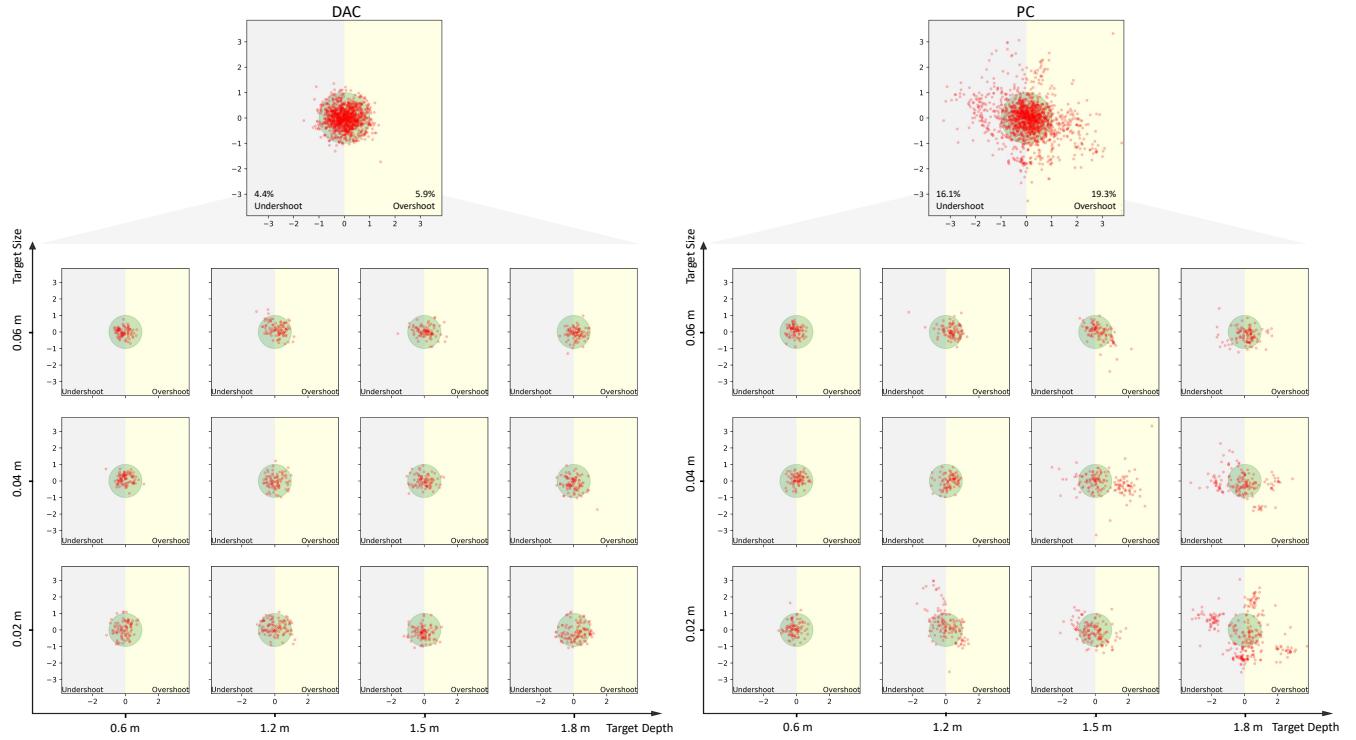


Figure 12: Normalized angular error for all selections collected in the 3D pointing task with (Left) DAC and (Right) PC. Each selection is represented as a red dot and the target is represented as a green circle in a normalized angular coordinate frame with the left and right half denoted as the undershooting and overshooting area respectively. Selections outside the target are erroneous clicks. These erroneous selections appear to be more scattered with higher angular error in PC than DAC, particularly at the lower right corner of the grid as *Target Depth* increases and *Target Size* decreases, indicating the effects of *Target Size* and *Target Depth* are more evident in PC than DAC by having more erroneous selections for small and distant targets.

would be important for applications that require precise selection of high ID objects, but may not be critical in tasks with imprecise selection. In practice, it is not common to see objects as small as 0.02 m in the state-of-the-art virtual desktop applications [16] that usually enlarge UI elements (buttons, toggles etc) for the ease of selection, possibly constrained by limited display resolutions. Therefore, applying the *Perspective Cursor* would be sufficient for current VR workspace applications, while depth-adaptation would be important for applications on high resolution HMDs that require precise selection of small objects.

5.2 Consistency with the Desktop Mouse

We found participants rated the consistency with desktop mouse positively in DAC over PC (Figure 10 (a)). In the interview, most participants (9 of 16) commented positively on the depth-adaptation behavior of DAC, for instance: “(P16) it was just like using the mouse normally”, “(P10) it kind of went in front of it (the object) and just automatically worked on it”, and “(P4) it felt much more like gliding with my two-dimensional movements to three-dimensional space because I sort of have this depth perception”. There are also three participants commenting negatively on the depth-adaptation. P1 felt fatigued: “I had to keep on wiggling my hand with the mouse, I can see my hands getting tired at some point”. P2 found it slow to

use: “I feel like I was probably slower because I had to slow down at the end to get into the right position”. P13 found it irritating when the cursor automatically moved depth-wise: “It has mouse moving depth-wise, which is irritating a bit”. We also collected participants’ performance on the 2D targets as a reference to their performance on the 3D targets. It is not surprising to see a similar performance on 2D targets with DAC and PC (Figure 10 (c)) as the two techniques work in the same way on 2D targets. The completion time of 3D targets in DAC is more consistent with the performance of the 2D targets group compared to PC. Overall, DAC provides better performance consistency between the 2D and 3D content than PC, and most participants liked its depth-adaptation.

The questionnaire result also shows participants’ ratings on the sensitivity of the cursors in the study compared to a regular desktop cursor (Figure 10 (a)). While there is no significant difference between the sensitivity of DAC and PC, the results tend to skew towards low overall, indicating that participants might prefer a higher CD gain. We used a constant CD gain of 38 DPI for all participants in a controlled study. In practice, users should be able to adjust its sensitivity in the usable range calculated from Table 1 based on their preferences. They can also use the usable range for pointer acceleration as prior studies [12, 14] have found that varying the CD gain based on input velocity improved user performance over

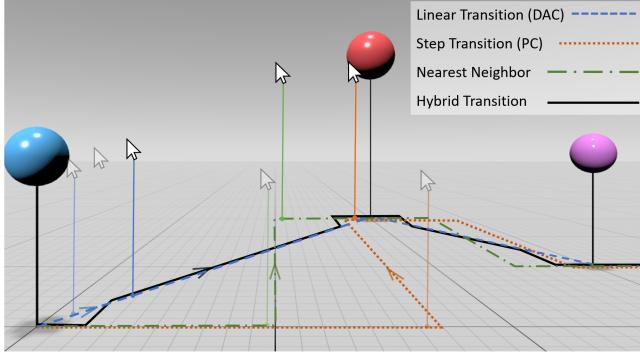


Figure 13: Different depth transition functions, including the step function in PC (orange dot line) with a snapping visual cue when hitting the target, the linear transition function in DAC (blue dash line) adapting linearly, and other depth functions that adapt to the depths of objects nearby.

constant CD gain in the desktop environment. In our work we used a constant CD gain, as prior work [52] has not found a significant improvement of using pointer acceleration in VR. However the study was designed for 2D pointing in VR with results contradicting previous findings in the desktop environment [12, 14]. Future experiments are required to compare pointer acceleration functions to constant CD gains for 3D pointing in VR.

5.3 Snapping Visual Cue

While most participants (9 of 16) commented negatively on PC, there are two participants mentioned that they liked the snapping behavior of PC because it provides a transitional visual cue indicating that the cursor is at the right spot ready to select: “(P1) it already tells me that I can click now”, and “(P2) there was this one point where there were like two circles really close to each other. If I was at a certain position close to it, the cursor would go there. I think in that situation it helped.” Participants might take the snapping visual cue as an indicator for target selection. This could potentially explain PC’s large variation of *Error Rate* in the 3D pointing task. Participants who used the visual cue would select targets once the cursor snapped onto the object, resulting in low *Error Rate*, while participants who did not use the cue made erroneous selections to get one-click hitting on the object. While this can be explained as a result of the speed-accuracy trade-off, we did not observe a similar large variation in DAC. Therefore we speculate the snapping visual cue of PC might contribute to the performance difference. Similarly in the data analysis task, *Error Rate* was even slightly lower in PC than DAC (Figure 9 (d)). There were four participants who made no erroneous selection in PC compared to two in DAC, indicating the snapping visual cue might be helpful to distinguish objects that are close to each other and varied in depth. The snapping visual cue can be used as an indicator of a cursor at a spot ready to select.

It should be noted that the snapping visual cue and DAC are not mutually exclusive. The depth-adaptation to the objects nearby does not have to be linear. We used the Laplacian interpolation that creates a linear transition between objects (blue dashed line in Figure 13). We used a step function in PC (orange dot line in

Figure 13) that makes the cursor snapping onto the object once the cursor reaches the object. Combining the step and linear function creates a hybrid transition between targets (black solid line in Figure 13) that can adapt to the targets nearby to avoid the *diplopia problem* while providing the snapping visual cue once the cursor is sufficiently close to the target. Reducing the linear transition in the hybrid approach leads to the nearest neighbor depth-adaptation (green dash-dot line in Figure 13), making it possible to integrate with *Bubble Cursor* [24] that used the nearest target to resize the cursor’s activation area to enhance target acquisition. There are a lot of possibilities for the depth-adaptation functions. While our work evaluated the linear transition, we hope future studies comparing different functions can find the optimal depth-adaptation function depending on different use cases.

5.4 Depth Map and AR HMDs

We generated the Voronoi diagram based on object center positions to interpolate the cursor depth. In practice, vertices on an object may have different depths relative to the viewpoint. Therefore, we used a simplified approach that takes the object center rather than its vertices. It works well with the small spheres used in our study. However, it should be noted that this approach will not provide depth continuity on objects with vertices that are varied in depth. Using the object center rather than the closest vertex to compute the cursor depth will cause the cursor to snap on its entry point to the object (Figure 14 (a)). Depending on the depth difference and the object’s visible size, this may re-surface the *diplopia problem*. For future work, we suggest using a depth map from the viewpoint to interpolate the cursor depth based on its potential entry point (the closest vertex) on each object (Figure 14 (b)).

In particular, the depth map can be useful for AR applications with a combination of physical and virtual objects that are varied in depth. A depth map with both virtual objects and physical surroundings can be used to interpolate the cursor depth so that the cursor can adapt to the selectable objects as well as the physical environment. The mouse cursor can also be used to interact with physical objects detected by AR HMDs such as adjusting speaker volume or turning on a light. We expect similar findings in AR HMDs that *Depth-Adaptive Cursor* can help users to precisely select small and distant objects. Future studies are required to investigate the characteristics of the mouse cursor in the AR workspace.

6 LIMITATIONS AND FUTURE WORK

In our study, we designed a 3D pointing task that encouraged users to alternate between a 2D and 3D target. We did not use the multi-directional tapping test (ISO 9241-9) in a circular setup [60] as we found it difficult to represent the typical use of the VR workspace which usually has 2D windows and 3D objects and therefore requires users to alternate between them. However, we acknowledge that our customized 3D pointing task makes it difficult to directly compare the results with other studies that used the standardized protocol. Therefore, we suggest that future work adapt a standardized method to include the alternation between 2D and 3D targets while preserving the capability to compare with other studies, such as placing half of the circular targets on a plane as 2D targets and the others in space as 3D targets. While we used a

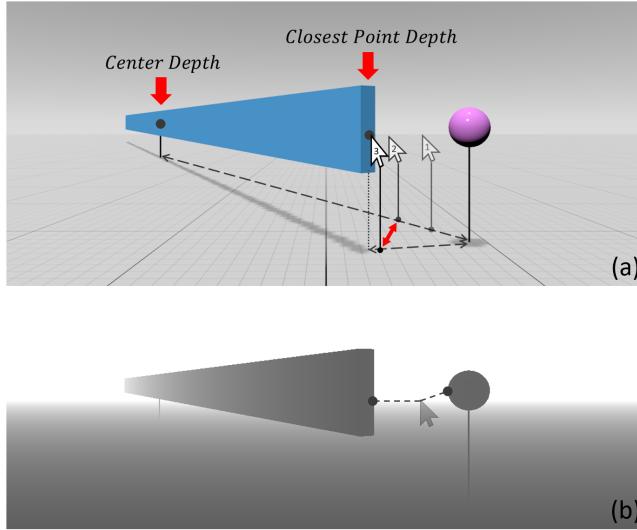


Figure 14: We suggest using the depth of closest point on an object as a prediction of the entry point to interpolate the cursor depth rather than the object center. (a) shows the depth difference between the center and closest point on a slant board varied in depth. Using the object center rather than the closest point to interpolate the cursor depth will cause the cursor to snap on its entry point to the board. (b) It can be avoided by finding the closest point for each object with a depth map from the viewpoint and using the closest points to interpolate the cursor depth.

different task setup, we did not observe participants performing extremely in the speed-accuracy tradeoff with the error rates of 7.1% (DAC) and 13.7% (PC), which are comparable to the error rate of about 10% in EZCursorVR [53] in the near depth that used standardized method for 2D pointing in VR. Future work with a standardized protocol adapted for the virtual workspace will be able to provide the throughput metric insensitive to the speed-accuracy tradeoff, which would facilitate comparison with other studies.

As we recruited existing HMD owners as participants, and a significantly greater proportion of HMD owners [29] are male, our participants were all male resulting in limited generalizability for all genders. Future work is required to investigate the potential effect of gender.

Prior work has found the desktop mouse to be less efficient in 3D manipulation tasks with high DoFs [7]. While our work focuses on selection, it would be interesting to explore how depth-adaptation applies to manipulation tasks such as translation and docking tasks. While it is possible to bring the standard 3D widgets from desktop to VR, it will be more interesting to investigate combined input modalities such as combining the mouse with mid-air gestures and eye-gaze so that the spatial input provides high DoFs while the mouse complements with stability and precision.

Our studies found that *Depth-Adaptive Cursor* enhances target selection with significant performance improvement over *Perspective Cursor*. It is most effective on small and distant targets. It is unclear how these findings apply to actual user interfaces in the

VR workspace. For future work, it would be interesting to evaluate it for target layouts resembling user interfaces of representative applications such as 3D modelling (Figure 1). While we evaluated *Depth-Adaptive Cursor* with static objects, it would be interesting to see how it works with dynamic objects that move in the 3D scene. Adapting to the moving objects will cause the cursor moving depth-wise, which will potentially affect users' performance.

7 CONCLUSION

With the increasing viability and appeal of conducting desktop knowledge work in VR, we examined the challenges of integrating a desktop mouse into the virtual workspace. These challenges include the *diplopia*, *perspective*, and *sensitivity problem*. To address these problems, we presented *Depth-Adaptive Cursor*, a 2D-mouse driven pointing technique for 3D objects with depth adaptation that continuously interpolates the cursor depth by inferring what users intend to select based on the cursor position, the viewpoint, and the selectable objects. It provides a control mechanism consistent with a regular desktop mouse. We also proposed a CD gain tool to compute a usable range of CD gains for general mouse-based pointing in VR HMDs. We conducted a user study to investigate the effectiveness of the depth-adaptation by comparing *Depth-Adaptive Cursor* to *Perspective Cursor* for 3D target selection in VR. Results showed that *Depth-Adaptive Cursor* significantly outperformed *Perspective Cursor* for target selection that requires high precision with reduced error, time, and workload. Together, our work investigated mice in-depth and explored the feasibility of integrating a desktop mouse into VR to support productive knowledge work.

ACKNOWLEDGMENTS

We thank our colleagues, Jo Vermeulen and Bon Aseniero for assistance in setting up the data analysis task, and Justin Matejka for feedback and suggestions in figures. We also thank the reviewers for insightful comments.

REFERENCES

- [1] Jonathan Aceituno, Géry Casiez, and Nicolas Roussel. 2013. How low can you go? Human limits in small unidirectional mouse movements. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1383–1386.
- [2] Tim Ameler, Kai Blohme, Lilith Brandt, Raphael Brüngel, Alice Hensel, Lisa Huber, Francis Kuper, Jessica Swoboda, Maren Warnecke, Michaela Warzecha, et al. 2019. A comparative evaluation of steamvr tracking and the optitrack system for medical device tracking. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 1465–1470.
- [3] Mark Ashdown, Kenji Oka, and Yoichi Sato. 2005. Combining head tracking and mouse input for a GUI on multiple monitors. In *CHI'05 extended abstracts on Human factors in computing systems*. 1188–1191.
- [4] Ravin Balakrishnan, Thomas Baudel, Gordon Kurtenbach, and George Fitzmaurice. 1997. The Rockin'Mouse: integral 3D manipulation on a plane. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. 311–318.
- [5] Patrick Baudisch, Edward Cutrell, Ken Hinckley, and Robert Gruen. 2004. Mouse ether: accelerating the acquisition of targets across multi-monitor displays. In *CHI'04 extended abstracts on Human factors in computing systems*. 1379–1382.
- [6] Hrvoje Benko and Steven Feiner. 2005. Multi-monitor mouse. In *CHI'05 extended abstracts on Human factors in computing systems*. 1208–1211.
- [7] François Bérard, Jessica Ip, Mitchel Benovoy, Dalia El-Shimy, Jeffrey R Blum, and Jeremy R Cooperstock. 2009. Did “Minority Report” get it wrong? Superiority of the mouse over 3D input devices in a 3D placement task. In *IFIP Conference on Human-Computer Interaction*. Springer, 400–414.
- [8] Lonní Besançon, Paul Issartel, Mehdi Ammi, and Tobias Isenberg. 2017. Mouse, tactile, and tangible input for 3D manipulation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 4727–4740.
- [9] Verena Biener, Daniel Schneider, Travis Gesslein, Alexander Otte, Bastian Kuth, Per Ola Kristensson, Eyal Ofek, Michel Pahud, and Jens Grubert. 2020. Breaking

- the Screen: Interaction Across Touchscreen Boundaries in Virtual Reality for Mobile Knowledge Workers. *IEEE Transactions on Visualization and Computer Graphics* 26, 12 (2020), 3490–3502.
- [10] Tom Bobach and Georg Umlauf. 2006. Natural neighbor interpolation and order of continuity. *GI Lecture Notes in Informatics: Visualization of Large and Unstructured Data Sets* (2006), 69–86.
- [11] Natalia Bogdan, Tovi Grossman, and George Fitzmaurice. 2014. HybridSpace: Integrating 3D freehand input and stereo viewing into traditional desktop applications. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 51–58.
- [12] Géry Casiez and Nicolas Roussel. 2011. No more bricolage! Methods and tools to characterize, replicate and compare pointing transfer functions. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*. 603–614.
- [13] Géry Casiez, Daniel Vogel, Ravin Balakrishnan, and Andy Cockburn. 2008. The impact of control-display gain on user performance in pointing tasks. *Human-computer interaction* 23, 3 (2008), 215–250.
- [14] Géry Casiez, Daniel Vogel, Qing Pan, and Christophe Chaillou. 2007. RubberEdge: reducing clutching by combining position and rate control with elastic feedback. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*. 129–138.
- [15] Mehmet Celal Dasiyici. 2008. *Multi-scale cursor: Optimizing mouse interaction for large personal workspaces*. Ph.D. Dissertation. Virginia Tech.
- [16] Virtual Desktop. 2021. Virtual Desktop: Your PC in VR. <https://www.vrdesktop.net/>. [Online accessed 8-August-2021].
- [17] Anna Eiberger, Per Ola Kristensson, Susanne Mayr, Matthias Kranz, and Jens Grubert. 2019. Effects of depth layer switching between an optical see-through head-mounted display and a body-proximate display. In *Symposium on Spatial User Interaction*. 1–9.
- [18] Alex Endert, Patrick Fiaux, Haeyoung Chung, Michael Stewart, Christopher Andrews, and Chris North. 2011. ChairMouse: leveraging natural chair rotation for cursor navigation on large, high-resolution displays. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. 571–580.
- [19] Yasin Farmani and Robert J Teather. 2017. Player performance with different input devices in virtual reality first-person shooter games.. In *SUI*. 165.
- [20] Andreas Rene Fender, Hrvoje Benko, and Andy Wilson. 2017. Meetalive: Room-scale omni-directional display system for multi-user content and control sharing. In *Proceedings of the 2017 ACM international conference on interactive surfaces and spaces*. 106–115.
- [21] Scott Frees and G Drew Kessler. 2005. Precise and rapid interaction through scaled manipulation in immersive virtual environments. In *IEEE Proceedings. VR 2005. Virtual Reality*. 2005. IEEE, 99–106.
- [22] Renaud Gervais, Jérémie Frey, and Martin Hatchet. 2015. Pointing in spatial augmented reality from 2D pointing devices. In *IFIP Conference on Human-Computer Interaction*. Springer, 381–389.
- [23] Renaud Gervais, Joan Sol Roo, and Martin Hatchet. 2016. Tangible viewports: Getting out of flatland in desktop environments. In *Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*. 176–184.
- [24] Tovi Grossman and Ravin Balakrishnan. 2005. 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*. 281–290.
- [25] J Grubert, E Ofek, M Pahud, and PO Kristensson. [n.d.]. Back to the Future: Revisiting Mouse and Keyboard Interaction for HMD-based Immersive Analytics. In *In ACM CHI 2020 4th Workshop on Immersive Analytics: Envisioning Future Productivity for Immersive Analytics*.
- [26] Jens Grubert, Eyal Ofek, Michel Pahud, and Per Ola Kristensson. 2018. The office of the future: Virtual, portable, and global. *IEEE computer graphics and applications* 38, 6 (2018), 125–133.
- [27] Jens Grubert, Lukas Witzani, Eyal Ofek, Michel Pahud, Matthias Kranz, and Per Ola Kristensson. 2018. Text entry in immersive head-mounted display-based virtual reality using standard keyboards. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 159–166.
- [28] Sandra G Hart and Lowell E Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*. Vol. 52. Elsevier, 139–183.
- [29] Jonathan W Kelly, Lucia A Cherep, Alex F Lim, Taylor Doty, and Stephen B Gilber. 2021. Who Are Virtual Reality Headset Owners? A Survey and Comparison of Headset Owners and Non-Owners. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 687–694.
- [30] Woojoo Kim, Junyoung Jung, and Shuping Xiong. 2020. VRMouse: Mouse emulation with the VR controller for 2D selection in VR. In *International Conference on Applied Human Factors and Ergonomics*. Springer, 663–670.
- [31] Regis Kopper, Doug A Bowman, Mara G Silva, and Ryan P McMahan. 2010. A human motor behavior model for distal pointing tasks. *International journal of human-computer studies* 68, 10 (2010), 603–615.
- [32] Kacper Kotulak, Adam Froń, Andrzej Krankowski, German Olivares Pulido, and Manuel Henrandez-Pajares. 2017. Sibsonian and non-Sibsonian natural neighbour interpolation of the total electron content value. *Acta Geophysica* 65, 1 (2017), 13–28.
- [33] Alena Kovarova and Maros Urbancok. 2014. Can virtual reality be better controlled by a smart phone than by a mouse and a keyboard?. In *Proceedings of the 15th International Conference on Computer Systems and Technologies*. 317–324.
- [34] Philipp Koytek, Charles Perin, Jo Vermeulen, Elisabeth Andre, and Sheelagh Carpendale. 2017. Mybrush: Brushing and linking with personal agency. *IEEE transactions on visualization and computer graphics* 24, 1 (2017), 605–615.
- [35] Max Krichenbauer, Goshiro Yamamoto, Takafumi Taketomi, Christian Sandor, and Hirokazu Kato. 2017. Augmented reality versus virtual reality for 3d object manipulation. *IEEE transactions on visualization and computer graphics* 24, 2 (2017), 1038–1048.
- [36] Joon Hyub Lee and Seok-Hyung Bae. 2013. Binocular cursor: enabling selection on transparent displays troubled by binocular parallax. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3169–3172.
- [37] Lenovo. 2021. ThinkReality A3 Smart Glasses Workspace. <https://www.lenovo.com/us/en/thinkrealitya3>. [Online accessed 8-August-2021].
- [38] Daniel Antonio Linares Garcia, Poorvesh Dongre, Nazila Roofigari-Esfahan, and Doug A Bowman. 2021. The Mobile Office: A Mobile AR Systems for Productivity Applications in Industrial Environments. In *International Conference on Human-Computer Interaction*. Springer, 511–532.
- [39] Lars Lischke, Valentin Schwind, Robin Schweigert, Paweł W Woźniak, and Niels Henze. 2019. Understanding pointing for workspace tasks on large high-resolution displays. In *Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia*. 1–9.
- [40] Mingyu Liu, Matthieu Nancel, and Daniel Vogel. 2015. Gunslinger: Subtle arms-down mid-air interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. 63–71.
- [41] Erin Martel and Kasia Muldner. 2017. Controlling VR games: control schemes and the player experience. *Entertainment computing* 21 (2017), 19–31.
- [42] M. McGuffin. 2019. Augmented Reality Knowledge Work : Towards a Research Agenda.
- [43] Brad A Myers, Rishi Bhatnagar, Jeffrey Nichols, Choon Hong Peck, Dave Kong, Robert Miller, and A Chris Long. 2002. Interacting at a distance: measuring the performance of laser pointers and other devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 33–40.
- [44] Miguel A Nacenta, Samer Sallam, Bernard Champoux, Sriram Subramanian, and Carl Gutwin. 2006. Perspective cursor: perspective-based interaction for multi-display environments. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 289–298.
- [45] Mathieu Nancel, Emmanuel Pietriga, Olivier Chapuis, and Michel Beaudouin-Lafon. 2015. Mid-air pointing on ultra-walls. *ACM Transactions on Computer-Human Interaction (TOCHI)* 22, 5 (2015), 1–62.
- [46] Oculus. 2021. Facebook Connect: Oculus Quest 2 Infinite Office. <https://www.oculus.com/blog/facebook-connect-oculus-quest-2-aaa-gaming-partnerships-and-more/>. [Online accessed 8-August-2021].
- [47] Oculus. 2021. Oculus Quest v25 Adds Resizable Browser Windows and Bluetooth Mouse Support. <https://uploadvr.com/oculus-quest-bluetooth-mouse/>. [Online accessed 8-August-2021].
- [48] Eyal Ofek, Jens Grubert, Michel Pahud, Mark Phillips, and Per Ola Kristensson. 2020. Towards a practical virtual office for mobile knowledge workers. *arXiv preprint arXiv:2009.02947* (2020).
- [49] Leonardo Pavattoni, Chris North, Doug A Bowman, Carmen Badea, and Richard Stoakley. 2021. Do we still need physical monitors? An evaluation of the usability of AR virtual monitors for productivity work. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 759–767.
- [50] Julian Petford, Miguel A Nacenta, and Carl Gutwin. 2018. 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*. 1–14.
- [51] Duc-Minh Pham and Wolfgang Stuerzlinger. 2019. Is the pen mightier than the controller? a comparison of input devices for selection in virtual and augmented reality. In *25th ACM Symposium on Virtual Reality Software and Technology*. 1–11.
- [52] Adrian Ramcharitar. 2018. *2D Selection in Virtual Reality with Head Mounted Displays*. Ph.D. Dissertation. Carleton University.
- [53] Adrian Ramcharitar and Robert J Teather. 2018. EZCursorVR: 2D selection with virtual reality head-mounted displays. In *Proceedings of the 44th Graphics Interface Conference*. 123–130.
- [54] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. 1998. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. 179–188.
- [55] Leila Schemali and Elmar Eisemann. 2014. Design and evaluation of mouse cursors in a stereoscopic desktop environment. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 67–70.
- [56] Daniel Schneider, Alexander Otte, Travis Gesslein, Philipp Gagel, Bastian Kuth, Mohamad Shahim Damlakhi, Oliver Dietz, Eyal Ofek, Michel Pahud, Per Ola Kristensson, et al. 2019. Reconfiguration: Reconfiguring physical keyboards in virtual reality. *IEEE transactions on visualization and computer graphics* 25, 11

- (2019), 3190–3201.
- [57] Jonmichael Seibert and Daniel M Shafer. 2018. Control mapping in virtual reality: effects on spatial presence and controller naturalness. *Virtual Reality* 22, 1 (2018), 79–88.
- [58] Marcos Serrano, Barrett Ens, Xing-Dong Yang, and Pourang Irani. 2015. Desktop-Gluey: Augmenting desktop environments with wearable devices. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*. 1175–1178.
- [59] Robert J Teather and Wolfgang Stuerzlinger. 2008. Assessing the Effects of Orientation and Device on 3D Positioning. In *2008 IEEE Virtual Reality Conference*. IEEE, 293–294.
- [60] 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*. 159–168.
- [61] Robert J Teather and Wolfgang Stuerzlinger. 2015. Factors affecting mouse-based 3d selection in desktop vr systems. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*. 10–19.
- [62] Varjo. 2021. Varjo Workspace. <https://varjo.com/use-center/get-to-know-your-headset/varjo-workspace/>. [Online accessed 8-August-2021].
- [63] Rafael Veras, Gaganpreet Singh, Farzin Farhadi-Niaki, Ritesh Udhani, Parth Pradeep Patekar, Wei Zhou, Pourang Irani, and Wei Li. 2021. Elbow-Anchored Interaction: Designing Restful Mid-Air Input. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.
- [64] Jorge A Wagner Filho, Carla Maria Dal Sasso Freitas, and Luciana Nedel. 2018. Virtualdesk: A comfortable and efficient immersive information visualization approach. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 415–426.
- [65] Manuela Waldner, Christian Pirchheim, Ernst Kruijff, and Dieter Schmalstieg. 2010. Automatic configuration of spatially consistent mouse pointer navigation in multi-display environments. In *Proceedings of the 15th international conference on Intelligent user interfaces*. 397–400.
- [66] Colin Ware and Kathy Lowther. 1997. Selection using a one-eyed cursor in a fish tank VR environment. *ACM Transactions on Computer-Human Interaction (TOCHI)* 4, 4 (1997), 309–322.
- [67] Wikipedia contributors. 2021. Comparison of virtual reality headsets — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Comparison_of_virtual_reality_headsets&oldid=1041684677. [Online; accessed 5-September-2021].
- [68] Robert Xiao, Miguel A Nacenta, Regan L Mandryk, Andy Cockburn, and Carl Gutwin. 2011. Ubiquitous cursor: a comparison of direct and indirect pointing feedback in multi-display environments. In *Proceedings of Graphics Interface 2011*. Citeseer, 135–142.